SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 1, pp. 1–19

BACKWARD PERTURBATION ANALYSIS OF THE PERIODIC DISCRETE-TIME ALGEBRAIC RICCATI EQUATION*

JI-GUANG SUN†

Abstract. Normwise backward errors and residual bounds for an approximate Hermitian positive semidefinite solution set to the periodic discrete-time algebraic Riccati equation are obtained. The results are illustrated by using simple numerical examples.

Key words. periodic Riccati equation, Hermitian positive semidefinite stabilizing solution set, backward error, residual bound

AMS subject classifications. 15A24, 65H10, 93B99

DOI. 10.1137/S0895479802414928

1. Introduction.

We consider the periodic discrete-time algebraic Riccati equation (P-DARE) with period $p \geq 2$:

$$\begin{aligned} X_{j-1} &= A_j^H X_j A_j - A_j^H X_j B_j (R_j + B_j^H X_j B_j)^{-1} B_j^H X_j A_j + H_j \\ &= A_j^H X_j (I + G_j X_j)^{-1} A_j + H_j, \end{aligned} \tag{1.1}$$

where, for all j, $A_j = A_{j+p}$, $H_j = H_{j+p}$ and $X_j = X_{j+p}$ are $n \times n$ matrices, $B_j = B_{j+p}$ are $n \times m$ matrices, and $R_j = R_{j+p}$ are $m \times m$ matrices; B_j is of full column rank, R_j is Hermitian positive definite ($R_j > 0$), $G_j = B_j R_j^{-1} B_j^H = G_{j+p}$, and H_j is Hermitian positive semidefinite (p.s.d.) with $H_j = C_j^H C_j$. Equation (1.1) arises frequently in solving periodic discrete-time linear optimal control problems [1], [2]. Appropriate assumptions on the coefficient matrices guarantee the existence and uniqueness of the Hermitian p.s.d. stabilizing solution set $\{X_j\}_{j=1}^p$ to the P-DARE (1.1) (see Theorem 2.5 of section 2). Note that for the case $p = 1$ we have a single Riccati equation for which backward perturbation bounds and residual bounds are known [15], [16].

A forward perturbation analysis of the P-DARE (1.1) is presented by Lin and Sun [12], where perturbation bounds and condition numbers of the Hermitian p.s.d. stabilizing solution set to the P-DARE are obtained [12, sections 3 and 4]. In this paper, we present a backward perturbation analysis of the P-DARE (1.1).

Backward perturbation analysis is motivated by the following fact. Let an approximate Hermitian p.s.d. solution set $\{\tilde{X}_j\}_{j=1}^p$ to the P-DARE (1.1) be given. For example, the approximate solution set may come from a numerical algorithm for approximating the exact Hermitian p.s.d. stabilizing solution set $\{X_j\}_{j=1}^p$. Then there are two questions associated with the approximate solution set: (1) Is the approximate solution set the exact solution set of a slightly perturbed P-DARE? (2) Is the approximate solution set close to the exact solution set $\{X_j\}_{j=1}^p$? The result of a backward perturbation analysis may be a backward error, or a residual bound. The purpose of backward perturbation analysis of the P-DARE (1.1) is to test the stability

*Received by the editors September 23, 2002; accepted for publication (in revised form) by V. Mehrmann July 22, 2003; published electronically August 6, 2004. This work was supported by the Swedish Strategic Research Foundation Grant entitled "Matrix Pencil Computations in Computer-Aided Control System Design: Theory, Algorithms and Software Tools."
http://www.siam.org/journals/simax/26-1/41492.html
†Department of Computing Science, Umeå University, S-901 87 Umeå, Sweden (jisun@cs.umu.se).

of a computation or an algorithm and to ascertain the accuracy of an approximate solution set.

In matrix computations, developing backward errors and residual bounds is a part of the subject of perturbation theory (see [7], [13], and [17]). In recent years, the study of backward errors and residual bounds of matrix equations has been developed considerably. Taking full account of the special structure of the Sylvester equation, Higham [6] evaluates the backward error of an approximate solution to the matrix equation and determines the sensitivity of the equation to perturbations in the data. After that, Kågström [9] evaluates the normwise backward error of an approximate solution to the generalized Sylvester equation, and determines the sensitivity of the equation; Ghavimi and Laub [4] present a new backward error criterion, together with a sensitivity measure, for assessing solution accuracy of nonsymmetric and symmetric continuous-time algebraic Riccati equations. Normwise backward errors and residual bounds for continuous-time and discrete-time algebraic Riccati equations are obtained by the author [14], [15], [16]. This work, as a generalization of the results given by [15] and [16], derives normwise backward errors and residual bounds for an approximate Hermitian p.s.d. solution set to the P-DARE (1.1).

We begin in section 2 with pd-stable matrices and the Hermitian p.s.d. stabilizing solution set to the P-DARE (1.1). In sections 3 and 4 we derive normwise backward errors and residual bounds for an approximate Hermitian p.s.d. solution set to the P-DARE (1.1), respectively. The results will be illustrated by simple numerical examples in section 5.

2. Preliminaries.

2.1. Notation. Throughout this paper, $\mathcal{C}_n$ and $\mathcal{H}_n$ denote the set of $n \times n$ complex and $n \times n$ Hermitian matrices, respectively, and $\mathcal{C}_n^p$ and $\mathcal{H}_n^p$ denote the p-tuple product spaces $\mathcal{C}_n \times \cdots \times \mathcal{C}_n$ and $\mathcal{H}_n \times \cdots \times \mathcal{H}_n$, respectively. $\overline{A}$ denotes the conjugate of a matrix A, A^T denotes the transpose of A, and $A^H = \overline{A}^T$. I stands for the identity matrix, I_n is the identity matrix of order n, and 0 is the null matrix. The set of all eigenvalues of A is denoted by $\lambda(A)$. The spectral radius $\rho(A)$ is defined by $\rho(A) = \max\{|\lambda_j| : \lambda_j \in \lambda(A)\}$. An $n \times n$ matrix Φ is said to be d-stable if $\rho(\Phi) < 1$. The symbol $\| \ \|_F$ is the Frobenius norm, and $\| \ \|_2$ is the spectral norm and the Euclidean vector norm. For $A = (a_1, \ldots, a_n) = (\alpha_{ij}) \in \mathcal{C}_n$ and a matrix B, $A \otimes B = (\alpha_{ij}B)$ is a Kronecker product, and vec$A$ is a vector defined by vec$A = (a_1^T, \ldots, a_n^T)^T$. For $A \in \mathcal{C}_n$ we have [5, pp. 32–34]

$$\text{vec}A^T = \Pi \text{vec}A,$$

where Π is the vec-permutation matrix which can be expressed by

$$\Pi = \sum_{k,l=1}^{n} e_k e_l^T \otimes e_l e_k^T,$$

in which e_k denotes the kth column of I_n. In order to save the space of the matrix representation, we use the following notation [12]:

$$\text{diag}\{N_j\}_{j=1}^p = \begin{pmatrix} N_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & N_p \end{pmatrix}, \quad \text{cyc}\{N_j\}_{j=1}^p = \begin{pmatrix} 0 & \cdots & 0 & N_1 \\ N_2 & \ddots & & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & N_p & 0 \end{pmatrix}.$$

2.2. On pd-stable matrices. We first cite some definitions from [12].

Let $\Phi_1, \ldots, \Phi_p \in \mathcal{C}_n$. If there are complex numbers $\alpha_1, \ldots, \alpha_p$ such that

$$\det\left[\mathrm{diag}\{\alpha_j I\}_{j=1}^p - \mathrm{cyc}\{\Phi_j\}_{j=1}^p\right] = 0,$$

then $\alpha_1 \cdots \alpha_p$ is an eigenvalue of the matrix set $\{\Phi_j\}_{j=1}^p$.

The set of all eigenvalues of $\{\Phi_j\}_{j=1}^p$ is denoted by $\lambda\left(\{\Phi_j\}_{j=1}^p\right)$. We have [12]

$$\lambda\left(\{\Phi_j\}_{j=1}^p\right) = \lambda(\Phi_p \Phi_{p-1} \cdots \Phi_1).$$

Consequently, if we define the spectral radius $\rho\left(\{\Phi_j\}_{j=1}^p\right)$ by

$$\rho\left(\{\Phi_j\}_{j=1}^p\right) = \max\left\{|\lambda_j| : \ \lambda_j \in \lambda\left(\{\Phi_j\}_{j=1}^p\right)\right\},$$

then

$$\rho\left(\{\Phi_j\}_{j=1}^p\right) = \rho(\Phi_p \Phi_{p-1} \cdots \Phi_1).$$

Let $\Phi_1, \ldots, \Phi_p \in \mathcal{C}_n$. The matrix p-tuple $\{\Phi_j\}_{j=1}^p$ is said to be pd-stable if the matrix $\Phi_p \Phi_{p-1} \cdots \Phi_1$ is d-stable.

Let $\Phi_1, \ldots, \Phi_p \in \mathcal{C}_n$. Define the linear operator $\mathbf{L} : \mathcal{H}_n^p \to \mathcal{H}_n^p$ by

(2.1)

$$\mathbf{L}(W_1, \ldots, W_p) = \left(W_1 - \Phi_2^H W_2 \Phi_2, \ldots, W_{p-1} - \Phi_p^H W_p \Phi_p, W_p - \Phi_1^H W_1 \Phi_1\right),$$

$$(W_1, \ldots, W_p) \in \mathcal{H}_n^p.$$

It is known [12] that the matrix L defined by

$$L = I_{pn^2} - \begin{pmatrix} 0 & \Phi_2^T \otimes \Phi_2^H & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \Phi_p^T \otimes \Phi_p^H \\ \Phi_1^T \otimes \Phi_1^H & \cdots & \cdots & 0 \end{pmatrix} \tag{2.2}$$

is a matrix representation of $\mathbf{L}$ on the space

$$\mathcal{H}^{pn^2} \equiv \left\{\left(w_1^T, \ldots, w_p^T\right)^T : \ w_j = \mathrm{vec} W_j, \ W_j \in \mathcal{H}_n \ \ \forall j\right\}.$$

LEMMA 2.1 (see [12, Lemma 2.1]). *The linear operator* $\mathbf{L}$ *defined by* (2.1) *is singular provided that there is an eigenvalue* $\lambda_k \in \lambda\left(\{\Phi_j\}_{j=1}^p\right)$ *with* $|\lambda_k| = 1$.

LEMMA 2.2 (see [12, Lemma 2.2]). *Let* $\Phi = \mathrm{cyc}\{\Phi_j\}_{j=1}^p$, *where* $\Phi_j \in \mathcal{C}_n \ \forall j$. *If* $\{\Phi_j\}_{j=1}^p$ *is pd-stable, then* Φ *is d-stable.*

Assume that $\{\Phi_j\}_{j=1}^p$ is pd-stable. By Lemma 2.2 the matrix L defined by (2.2) is nonsingular, and thus $\mathbf{L}^{-1}$ exists. In such a case, we define the quantity l by

$$l = \|\mathbf{L}^{-1}\|^{-1},$$

where the operator norm $\|\ \|$ for $\mathbf{L}^{-1}$ is induced by the Frobenius norm $\|\ \|_F$ on $\mathcal{H}_n^p$. Note that $\mathcal{H}_n^p$ is not a subspace of $\mathcal{C}_n^p$, but by [12, Appendix (I)] we have

$$l = \|L^{-1}\|_2^{-1}, \tag{2.3}$$

i.e., the induced operator norms of $\mathbf{L}^{-1}$ on $\mathcal{C}_n^p$ and $\mathcal{H}_n^p$ are equal.

Let the matrix set $\{\Phi_j\}_{j=1}^p$ be pd-stable. Define

$$s_{\rm pd} = \min\left\{ \max_{1\le j\le p} \|E_j\|_2 :\ \rho\left(\left\{(I-E_j)^{-1}\Phi_j\right\}_{j=1}^p\right) = 1,\ E_j\in\mathcal{C}_n\ \ \forall j \right\}. \tag{2.4}$$

The quantity $s_{\rm pd}$ measures the smallest $\max_{1\le j\le p}\|E_j\|_2$ such that $\left\{(I-E_j)^{-1}\Phi_j\right\}_{j=1}^p$ has an eigenvalue on the unit circle. Note that the computation of $s_{\rm pd}$ may be a rather difficult computational problem in the general case.

LEMMA 2.3. *Let $\{\Phi_j\}_{j=1}^p$ be pd-stable, and let $\mathbf{L}$ be the linear operator defined by* (2.1) *with L of* (2.2) *as its matrix representation. Let l and $s_{\rm pd}$ be the quantities defined by* (2.3) *and* (2.4)*, respectively, and let*

$$\phi_j = \|\Phi_j\|_2, \qquad \phi = \max_{1\le j\le p}\phi_j. \tag{2.5}$$

Then

$$\frac{l}{\phi^2+\phi\sqrt{\phi^2+l}+l} \le s_{\rm pd}. \tag{2.6}$$

Proof. Let the matrices $E_j^*\in\mathcal{C}_n$ $(j=1,\dots,p)$ satisfy

$$s_{\rm pd} = \max_{1\le j\le p}\|E_j^*\|_2 \quad\text{with}\quad \rho\left(\left\{(I-E_j^*)^{-1}\Phi_j\right\}_{j=1}^p\right) = 1.$$

By Lemma 2.1 the transformation

$$\begin{pmatrix} W_1 \\ \vdots \\ W_{p-1} \\ W_p \end{pmatrix} \mapsto \begin{pmatrix} W_1 - \left[(I-E_2^*)^{-1}\Phi_2\right]^H W_2 \left[(I-E_2^*)^{-1}\Phi_2\right] \\ \vdots \\ W_{p-1} - \left[\left(I-E_p^*\right)^{-1}\Phi_p\right]^H W_p \left[\left(I-E_p^*\right)^{-1}\Phi_p\right] \\ W_p - \left[(I-E_1^*)^{-1}\Phi_1\right]^H W_1 \left[(I-E_1^*)^{-1}\Phi_1\right] \end{pmatrix}$$

is singular, where $W_j\in\mathcal{H}_n$ for all j; i.e., there are Hermitian matrices $W_1^*,\dots,W_p^*$ with $W_k\ne 0$ for some index $k\in\{1,\dots,p\}$ such that

$$\begin{aligned} W_1^* - \left[(I-E_2^*)^{-1}\Phi_2\right]^H W_2^* \left[(I-E_2^*)^{-1}\Phi_2\right] &= 0, \\ &\vdots \\ W_p^* - \left[(I-E_1^*)^{-1}\Phi_1\right]^H W_1^* \left[(I-E_1^*)^{-1}\Phi_1\right] &= 0. \end{aligned} \tag{2.7}$$

Let $N_j\in\mathcal{C}_n$ be defined by

$$I + N_j = \left(I-E_j^*\right)^{-1}, \quad j=1,\dots,p. \tag{2.8}$$

Then (2.7) can be written as

$$\mathbf{L}\begin{pmatrix} W_1^{*T} \\ \vdots \\ W_p^{*T} \end{pmatrix}^T = \begin{pmatrix} \left(\Phi_2^H W_2^* N_2\Phi_2 + \Phi_2^H N_2^H W_2^*\Phi_2 + \Phi_2^H N_2^H W_2^* N_2\Phi_2\right)^T \\ \vdots \\ \left(\Phi_1^H W_1^* N_1\Phi_1 + \Phi_1^H N_1^H W_1^*\Phi_1 + \Phi_1^H N_1^H W_1^* N_1\Phi_1\right)^T \end{pmatrix}^T, \tag{2.9}$$

or equivalently, by letting $\mathrm{vec}W_j^* = w_j^*$ $(j = 1, \ldots, p)$, we have

$$L \begin{pmatrix} w_1^* \\ \vdots \\ w_p^* \end{pmatrix} = \left(\mathrm{cyc}\left\{\Omega_j^T\right\}_{j=1}^p\right)^T \begin{pmatrix} w_1^* \\ \vdots \\ w_p^* \end{pmatrix}, \tag{2.10}$$

where

$$\Omega_j = \left(\Phi_j^T N_j^T\right) \otimes \Phi_j^H + \Phi_j^T \otimes \left(\Phi_j^H N_j^H\right) + \left(\Phi_j^T N_j^T\right) \otimes \left(\Phi_j^H N_j^H\right) \quad \forall j.$$

Inverting L and taking 2-norm of the two sides of (2.10) we get

$$\nu^2 + 2\nu - \frac{l}{\phi^2} \geq 0, \tag{2.11}$$

where

$$\nu = \max_{1 \leq j \leq p} \nu_j \quad \text{with} \quad \nu_j = \|N_j\|_2 \ \ \forall j, \tag{2.12}$$

and by (2.11),

$$\nu \geq \sqrt{1 + \frac{l}{\phi^2}} - 1 = \frac{l}{\phi^2 + \phi\sqrt{\phi^2 + l}}. \tag{2.13}$$

Observe that the relations of (2.8) imply

$$N_j = (I + N_j)E_j^*$$

and

$$\|N_j\|_2 \leq (1 + \|N_j\|_2)\|E_j^*\|_2.$$

Hence, we have

$$\|E_j^*\|_2 \geq \frac{\|N_j\|_2}{1 + \|N_j\|_2},$$

and

$$\max_{1 \leq j \leq p} \|E_j^*\|_2 \geq \frac{\max\limits_{1 \leq j \leq p} \|N_j\|_2}{1 + \max\limits_{1 \leq j \leq p} \|N_j\|_2} = \frac{\nu}{1 + \nu}.$$

Combining it with (2.13) gives the inequality (2.6). ∎

From Lemma 2.3 we get the following lemma.

LEMMA 2.4. *Let $\{\Phi_j\}_{j=1}^p$ be pd-stable, and let $\mathbf{L}$ be the linear operator defined by* (2.1) *with L in* (2.2) *as its matrix representation. Moreover, let l and ϕ be defined by* (2.3) *and* (2.5)*, respectively. If $E_j \in \mathcal{C}_n$ $(j = 1, \ldots, p)$ satisfy*

$$\max_{1 \leq j \leq p} \|E_j\|_2 < \frac{l}{\phi^2 + \phi\sqrt{\phi^2 + l} + l},$$

then the matrix set $\{(\Phi_j + E_j)\}_{j=1}^p$ is pd-stable.

2.3. The Hermitian p.s.d. stabilizing solution set. By [2], the matrix pair sets $\{(A_j, B_j)\}_{j=1}^p$ and $\{(A_j, C_j)\}_{j=1}^p$ are said to be pd-stabilizable and pd-detectable, respectively, if the pairs $(\mathcal{A}_j, \mathcal{B}_j)$ and $(\mathcal{A}_j, \mathcal{C}_j)$ are d-stabilizable and d-detectable, respectively, for $j = 1, \ldots, p$, where

(2.14)

$$\mathcal{A}_j = A_{\pi_j(p)} \cdots A_{\pi_j(1)},$$

$$\mathcal{B}_j = (A_{\pi_j(p)} \cdots A_{\pi_j(2)} B_{\pi_j(1)}, A_{\pi_j(p)} \cdots A_{\pi_j(3)} B_{\pi_j(2)}, \ldots, A_{\pi_j(p)} B_{\pi_j(p-1)}, B_{\pi_j(p)}),$$

$$\mathcal{C}_j = \left(C^T_{\pi_j(1)}, A^T_{\pi_j(1)} C^T_{\pi_j(2)}, A^T_{\pi_j(1)} A^T_{\pi_j(2)} C^T_{\pi_j(3)}, \ldots, A^T_{\pi_j(1)} \cdots A^T_{\pi_j(p-1)} C^T_{\pi_j(p)}\right)^T,$$

and $\pi_j(\ )$ is a permutation defined by

$$\pi_j(k) = \begin{cases} k - j + 1 + p & \text{for } k = 1, \ldots, j-1 \text{ and } j \geq 2, \\ k - j + 1 & \text{for } k = j, \ldots, p. \end{cases} \tag{2.15}$$

Note that the pair (A, B) is d-stabilizable if $w^H B = 0$ and $w^H A = \lambda w^H$ for some constant λ implies $|\lambda| < 1$ or $w = 0$, and that the pair (A, C) is d-detectable if (A^H, C^H) is d-stabilizable.

Let $X_j \in \mathcal{H}_n$ $(j = 1, \ldots, p)$ and $\{X_j\}_{j=1}^p$ be a solution set to the P-DARE (1.1). If the matrix set $\{(I + G_j X_j)^{-1} A_j\}_{j=1}^p$ is pd-stable, then $\{X_j\}_{j=1}^p$ is said to be a stabilizing solution set to (1.1). If $X_j \geq 0$ for all j, then $\{X_j\}_{j=1}^p$ is said to be a Hermitian p.s.d. solution set.

The following result is a basic result on the existence and uniqueness of Hermitian p.s.d. stabilizing solution sets to the P-DARE (1.1). (See [1], [2], [12].)

THEOREM 2.5. *For the P-DARE* (1.1)*, if* $\{(A_j, B_j)\}_{j=1}^p$ *and* $\{(A_j, C_j)\}_{j=1}^p$ *are pd-stabilizable and pd-detectable, respectively, then there is a unique Hermitian p.s.d. stabilizing solution set* $\{X_j\}_{j=1}^p$ *to the P-DARE* (1.1)*.*

The result will be illustrated by Example 5.1 of section 5.

Throughout this paper, the matrix pair sets $\{(A_j, B_j)\}_{j=1}^p$ and $\{(A_j, C_j)\}_{j=1}^p$ of (1.1) are assumed to be pd-stabilizable and pd-detectable, respectively.

3. Backward errors.

3.1. Definitions. Let $\{\tilde{X}_j\}_{j=1}^p$ approximate the unique Hermitian p.s.d. stabilizing solution set to the P-DARE (1.1), and assume that the matrices $I + G_j \tilde{X}_j$ $(j = 1, \ldots, p)$ are nonsingular. Moreover, let $\Delta A_j, \Delta G_j, \Delta H_j$ be the corresponding perturbations in the coefficient matrices A_j, G_j, H_j $(j = 1, \ldots, p)$ of (1.1), respectively. The normwise backward error $\eta(\{\tilde{X}_j\}_{j=1}^p)$ of the approximate solution set $\{\tilde{X}_j\}_{j=1}^p$ can be defined by

$$\eta(\{\tilde{X}_j\}_{j=1}^p) = \max_{1 \leq j \leq p} \min_{\{(\Delta A_j, \Delta G_j, \Delta H_j)\}_{j=1}^p \in \mathcal{E}} \left\| \left(\frac{\Delta A_j}{\alpha_j}, \frac{\Delta G_j}{\beta_j}, \frac{\Delta H_j}{\gamma_j} \right) \right\|_F, \tag{3.1}$$

where the set $\mathcal{E}$ is defined by

$$\mathcal{E} = \left\{ \{(\Delta A_j, \Delta G_j, \Delta H_j)\}_{j=1}^{p} : \begin{array}{l} \Delta A_j \in \mathcal{C}_n,\ \Delta G_j, \Delta H_j \in \mathcal{H}_n, \\ \tilde{X}_{j-1} \\ = (A_j + \Delta A_j)^H \tilde{X}_j \left[I + (G_j + \Delta G_j)\tilde{X}_j \right]^{-1} (A_j + \Delta A_j) \\ \quad + H_j + \Delta H_j, \\ j = 1, \ldots, p \end{array} \right\}, \tag{3.2}$$

and $\alpha_j, \beta_j, \gamma_j$ $(j = 1, \ldots, p)$ are positive parameters. Taking $\alpha_j = \beta_j = \gamma_j = 1$ for $j = 1, \ldots, p$ yields the normwise absolute backward error $\eta_{\mathrm{abs}}(\{\tilde{X}_j\}_{j=1}^{p})$, and taking $\alpha_j = \|A_j\|_F, \beta_j = \|G_j\|_F, \gamma_j = \|H_j\|_F$ $(j = 1, \ldots, p)$ yields the normwise relative backward error $\eta_{\mathrm{rel}}(\{\tilde{X}_j\}_{j=1}^{p})$.

From (3.1) and (3.2) we see that the backward error $\eta(\{\tilde{X}_j\}_{j=1}^{p})$ of an approximate Hermitian solution set $\{\tilde{X}_j\}_{j=1}^{p}$ to the P-DARE (1.1) is a measure of "smallest" perturbations $\Delta A_j/\alpha_j, \Delta G_j/\beta_j, \Delta H_j/\gamma_j$ $(j = 1, \ldots, p)$ such that $\{\tilde{X}_j\}_{j=1}^{p}$ is just a Hermitian solution set to the perturbed P-DARE

$$\begin{aligned} \tilde{X}_{j-1} = (A_j + \Delta A_j)^H \tilde{X}_j [I + (G_j + \Delta G_j)\tilde{X}_j]^{-1}(A_j + \Delta A_j) + H_j + \Delta H_j, \\ j = 1, \ldots, p. \end{aligned} \tag{3.3}$$

Moreover, from (3.1) and (3.2) we see that

$$\eta(\{\tilde{X}_j\}_{j=1}^{p}) = \max_{1 \le j \le p} \eta_j, \tag{3.4}$$

where each η_j is defined by

$$\eta_j = \min_{(\Delta A_j, \Delta G_j, \Delta H_j) \in \mathcal{E}_j} \left\| \left(\frac{\Delta A_j}{\alpha_j}, \frac{\Delta G_j}{\beta_j}, \frac{\Delta H_j}{\gamma_j} \right) \right\|_F, \tag{3.5}$$

in which the set $\mathcal{E}_j$ is defined by

$$\mathcal{E}_j = \left\{ (\Delta A_j, \Delta G_j, \Delta H_j) : \begin{array}{l} \Delta A_j \in \mathcal{C}_n,\ \Delta G_j, \Delta H_j \in \mathcal{H}_n, \\ \tilde{X}_{j-1} \\ = (A_j + \Delta A_j)^H \tilde{X}_j \left[I + (G_j + \Delta G_j)\tilde{X}_j \right]^{-1} (A_j + \Delta A_j) \\ \quad + H_j + \Delta H_j \end{array} \right\}. \tag{3.6}$$

Consequently, the problem of estimating the backward error $\eta(\{\tilde{X}_j\}_{j=1}^{p})$ is reduced to the problem of estimating η_j for $j = 1, \ldots, p$.

3.2. Estimates of η_j $(j = 1, \ldots, p)$. For each $j \in \{1, \ldots, p\}$ define

$$\tilde{L}_j = \tilde{X}_j(I + G_j\tilde{X}_j)^{-1} \in \mathcal{H}_n, \quad \tilde{K}_j = \tilde{L}_j A_j \in \mathcal{C}_n, \tag{3.7}$$

and define the residual $\widehat{R}_j$ by

$$\widehat{R}_j = \tilde{X}_{j-1} - A_j^H\tilde{X}_j(I + G_j\tilde{X}_j)^{-1}A_j - H_j, \tag{3.8}$$

where $\tilde{X}_0 = \tilde{X}_p$. Moreover, define

$$\begin{aligned} &q_j(\Delta A_j, \Delta G_j) \\ &= -\tilde{K}_j^H\Delta G_j\tilde{L}_j\Delta G_j(I + \tilde{L}_j\Delta G_j)^{-1}\tilde{K}_j + \tilde{K}_j^H\Delta G_j(I + \tilde{L}_j\Delta G_j)^{-1}\tilde{L}_j\Delta A_j \\ &\quad +\Delta A_j^H\tilde{L}_j\Delta G_j(I + \tilde{L}_j\Delta G_j)^{-1}\tilde{K}_j - \Delta A_j^H(I + \tilde{L}_j\Delta G_j)^{-1}\tilde{L}_j\Delta A_j. \end{aligned} \tag{3.9}$$

Then by [15, section 2], the jth equation of (3.3) is equivalent to

$$\tilde{K}_j^H\Delta A_j + \Delta A_j^H\tilde{K}_j - \tilde{K}_j^H\Delta G_j\tilde{K}_j + \Delta H_j = \widehat{R}_j + q_j(\Delta A_j, \Delta G_j). \tag{3.10}$$

3.2.1. The real case. We now consider the case that all the coefficient matrices A_j, G_j, H_j; the perturbations $\Delta A_j, \Delta G_j, \Delta H_j$; and the approximate solution set $\{\tilde{X}_j\}_{j=1}^p$ are real. In such a case, (3.10) can be written as

$$\tilde{K}_j^T\Delta A_j + \Delta A_j^T\tilde{K}_j - \tilde{K}_j^T\Delta G_j\tilde{K}_j + \Delta H_j = \widehat{R}_j + q_j(\Delta A_j, \Delta G_j). \tag{3.11}$$

Define the matrix T_j by

$$T_j = \left(\alpha_j\left[I_n \otimes \tilde{K}_j^T + \left(\tilde{K}_j^T \otimes I_n\right)\Pi\right],\ -\beta_j\tilde{K}_j^T \otimes \tilde{K}_j^T,\ \gamma_j I_{n^2}\right), \tag{3.12}$$

where Π is the vec-permutation matrix. Then (3.11) is equivalent to the nonlinear system

$$T_j\begin{pmatrix} \dfrac{\mathrm{vec}\Delta A_j}{\alpha_j} \\ \dfrac{\mathrm{vec}\Delta G_j}{\beta_j} \\ \dfrac{\mathrm{vec}\Delta H_j}{\gamma_j} \end{pmatrix} = \mathrm{vec}\widehat{R}_j + \mathrm{vec}q_j(\Delta A_j, \Delta G_j). \tag{3.13}$$

By using the technique described by [15, section 2] we can prove the following result.

THEOREM 3.1. *For each $j \in \{1, \ldots, p\}$, let T_j be the matrix defined by* (3.12)*, and define τ_j, ρ_j, μ_j, and ν_j by*

$$\begin{aligned} &\tau_j = \left\|T_j^\dagger\right\|_2, \qquad \rho_j = \left\|T_j^\dagger\mathrm{vec}\widehat{R}_j\right\|_2, \\ &\mu_j = \left(\alpha_j^2 + \beta_j^2\|\tilde{K}_j\|_2^2\right)\|\tilde{L}\|_2, \quad \nu_j = \beta_j\|\tilde{X}_j\|_2\left\|(I + G_j\tilde{X}_j)^{-1}\right\|_2, \end{aligned} \tag{3.14}$$

where $T_j^{\dagger}$ *denotes the Moore–Penrose inverse of* T_j*, and* $\tilde{L}_j, \tilde{K}_j$*, and* $\widehat{R}_j$ *are the matrices defined by* (3.7) *and* (3.8)*. If*

$$\rho_j \le \min\left\{\frac{1}{\nu_j},\ \frac{\tau_j}{\tau_j\nu_j + 2\mu_j + \sqrt{(\tau_j\nu_j+2\mu_j)^2 - \tau_j^2\nu_j^2}}\right\}, \tag{3.15}$$

then

$$l_j \le \eta_j \le u_j, \tag{3.16}$$

where

$$\begin{aligned} u_j &= \frac{2\tau_j\rho_j}{\tau_j(1+\nu_j\rho_j) + \sqrt{\tau_j^2(1+\nu_j\rho_j)^2 - 4\tau_j(\tau_j\nu_j+\mu_j)\rho_j}}, \\ l_j &= \rho_j - \frac{\mu_j u_j^2}{\tau_j(1-\nu_j u_j)}. \end{aligned} \tag{3.17}$$

From Theorem 3.1 and the relation (3.4) we get the nonlinear estimates

$$l^* \equiv \max_{1\le j\le p} l_j \le \eta(\{\tilde{X}_j\}_{j=1}^p) \le \max_{1\le j\le p} u_j \equiv u^*. \tag{3.18}$$

Note that

$$u_j = \rho_j + \frac{\mu_j}{\tau_j}\rho_j^2 + O(\rho^3), \quad l_j = \rho_j - \frac{\mu_j}{\tau_j}\rho_j^2 + O(\rho^3), \quad j = 1,\dots,p.$$

Consequently, we have the linear estimates

$$\eta_j \approx \rho_j \ \ \forall j, \quad \text{and} \quad \eta(\{\tilde{X}_j\}_{j=1}^p) \approx \max_{1\le j\le p}\rho_j \tag{3.19}$$

as $\max_{1\le j\le p}\rho_j \to 0$ $(j\to\infty)$.

3.2.2. The complex case. Let

$$\begin{aligned} &\alpha_j\left[I_n\otimes\tilde{K}_j^H + (\tilde{K}_j^T\otimes I_n)\Pi\right] = U_{j,1} + i\Omega_{j,1}, \\ &-\beta_j\tilde{K}^T\otimes\tilde{K}_j^H = U_{j,2} + i\Omega_{j,2}, \\ &\mathrm{vec}\Delta A_j = x_j + iy_j, \quad \mathrm{vec}\Delta G_j = u_j + iv_j, \quad \mathrm{vec}\Delta H_j = z_j + iw_j, \\ &\mathrm{vec}\widehat{R}_j = r_j + is_j, \quad \mathrm{vec}q_j(\Delta A_j, \Delta G_j) = a_j + ib_j, \quad i = \sqrt{-1}, \end{aligned}$$

where $U_{j,k}$ and $\Omega_{j,k}$ $(k=1,2)$ are real matrices, and $x_j, y_j, u_j, v_j, z_j, w_j, r_j, s_j, a_j, b_j$ are real vectors. Moreover, let

$$T_j^{(c)} = \begin{pmatrix} U_{j,1} & -\Omega_{j,1} & U_{j,2} & -\Omega_{j,2} & \gamma_j I_{n^2} & 0 \\ \Omega_{j,1} & U_{j,1} & \Omega_{j,2} & U_{j,2} & 0 & \gamma_j I_{n^2} \end{pmatrix}, \tag{3.20}$$

and

$$\chi_j = \left(\frac{x_j^T}{\alpha_j},\ \frac{y_j^T}{\alpha_j},\ \frac{u_j^T}{\beta_j},\ \frac{v_j^T}{\beta_j},\ \frac{z_j^T}{\gamma_j},\ \frac{w_j^T}{\gamma_j} \right)^T.$$

Then (3.10) is equivalent to

$$T_j^{(c)} \chi_j = \begin{pmatrix} r_j \\ s_j \end{pmatrix} + \begin{pmatrix} a_j \\ b_j \end{pmatrix}.$$

Referring to [10], [11], and the proof of Theorem 3.1, we can prove the following result.

THEOREM 3.2. *For each $j \in \{1, \ldots, p\}$, let $T_j^{(c)}$ be the matrix defined by* (3.20). *Define μ_j and ν_j by* (3.14), *and define $\tau_j^{(c)}$ and $\rho_j^{(c)}$ by*

$$\tau_j^{(c)} = \left\| T_j^{(c)\dagger} \right\|_2, \quad \rho_j^{(c)} = \left\| T_j^{(c)\dagger} \begin{pmatrix} r_j \\ s_j \end{pmatrix} \right\|_2.$$

If

$$\rho_j^{(c)} \le \min \left\{ \frac{1}{\nu_j},\ \frac{\tau_j^{(c)}}{\tau_j^{(c)} \nu_j + 2\mu_j + \sqrt{(\tau_j^{(c)} \nu_j + 2\mu_j)^2 - {\tau_j^{(c)}}^2 \nu_j^2}} \right\},$$

then we have

$$l_j^{(c)} \le \eta_j \le u_j^{(c)},$$

where

$$u_j^{(c)} = \frac{2\tau_j^{(c)} \rho_j^{(c)}}{\tau_j^{(c)}(1 + \nu_j \rho_j^{(c)}) + \sqrt{{\tau_j^{(c)}}^2 (1 + \nu_j \rho_j^{(c)})^2 - 4\tau_j^{(c)} (\tau_j^{(c)} \nu_j + \mu_j) \rho_j^{(c)}}},$$

$$l_j^{(c)} = \rho_j^{(c)} - \frac{\mu_j {u_j^{(c)}}^2}{\tau_j^{(c)}(1 - \nu_j u_j^{(c)})}.$$

From Theorem 3.2 and the relation (3.4) we get

$$l^{(c)} \equiv \max_{1 \le j \le p} l_j^{(c)} \le \eta(\{\tilde{X}_j\}_{j=1}^p) \le \max_{1 \le j \le p} u_j^{(c)} \equiv u^{(c)}.$$

4. Residual bounds. In this section we prove the following result.

THEOREM 4.1. *Let $\{\tilde{X}_j\}_{j=1}^p$ be an approximate Hermitian solution set to the P-DARE* (1.1) *such that the matrices $I + G_j \tilde{X}_j$ $(j = 1, \ldots, p)$ are nonsingular, and the matrix set $\{(I + G_j \tilde{X}_j)^{-1} A_j\}_{j=1}^p$ is pd-stable. Define the residuals $\widehat{R}_j$ by*

$$\widehat{R}_j = \tilde{X}_{j-1} - A_j^H \tilde{X}_j (I + G_j \tilde{X}_j)^{-1} A_j - H_j, \quad j = 1, \ldots, p, \tag{4.1}$$

where $\tilde{X}_0 = \tilde{X}_p$, and define the linear operator $\mathbf{L} : \mathcal{H}_n^p \to \mathcal{H}_n^p$ by

$$\mathbf{L}(W_1, \ldots, W_{p-1}, W_p) = \left(W_1 - \tilde{\Phi}_2^H W_2 \tilde{\Phi}_2, \ldots, W_{p-1} - \tilde{\Phi}_p^H W_p \tilde{\Phi}_p, W_p - \tilde{\Phi}_1^H W_1 \tilde{\Phi}_1 \right), \tag{4.2}$$

where $W_1, \ldots, W_p \in \mathcal{H}_n$, *and the matrices* $\tilde{\Phi}_j$ *are defined by*

$$\tilde{\Phi}_j = (I + G_j \tilde{X}_j)^{-1} A_j, \quad j = 1, \ldots, p. \tag{4.3}$$

Moreover, let

$$\begin{aligned} &\phi = \max_{1\le j\le p} \phi_j \quad \text{with} \quad \phi_j = \|\tilde{\Phi}_j\|_2 \ \ \forall j, \\ &\gamma = \max_{1\le j\le p} \gamma_j \quad \text{with} \quad \gamma_j = \left\|(I + G_j \tilde{X}_j)^{-1} G_j\right\|_2 \ \ \forall j, \end{aligned} \tag{4.4}$$

and

$$l = \left\|\mathbf{L}^{-1}\right\|^{-1}, \qquad \epsilon = \left\|\mathbf{L}^{-1}(\widehat{R}_1, \ldots, \widehat{R}_p)\right\|_F, \tag{4.5}$$

where the operator norm $\|\ \|$ *for* $\mathbf{L}^{-1}$ *is induced by the Frobenius norm* $\|\ \|_F$ *on* $\mathcal{C}_n^p$. *If*

$$\epsilon < \frac{l}{\gamma(2\phi^2 + 2\phi\sqrt{\phi^2 + l} + l)}, \tag{4.6}$$

then for the unique Hermitian p.s.d. stabilizing solution set $\{X_j\}_{j=1}^p$ *to the P-DARE* (1.1) *we have*

$$\left\|(\tilde{X}_1 - X_1, \ldots, \tilde{X}_p - X_p)\right\|_F \le \frac{2l\epsilon}{(1+\gamma\epsilon)l + \sqrt{(1+\gamma\epsilon)^2 l^2 - 4(\phi^2 + l)\gamma l\epsilon}} \equiv r(\epsilon). \tag{4.7}$$

As a corollary of Theorem 4.1, we have the estimate

$$\left\|(\tilde{X}_1 - X_1, \ldots, \tilde{X}_p - X_p)\right\|_F \le \frac{2\epsilon}{1+\gamma\epsilon} = \frac{2\left\|\mathbf{L}^{-1}(\widehat{R}_1, \ldots, \widehat{R}_p)\right\|_F}{1 + \gamma\left\|\mathbf{L}^{-1}(\widehat{R}_1, \ldots, \widehat{R}_p)\right\|_F}.$$

Moreover, from (4.7) we obtain a relative error bound $b_{\mathrm{rel}}(\tilde{X}_j)$ for each $\tilde{X}_j$ ($1 \le j \le p$):

$$\frac{\|\tilde{X}_j - X_j\|_F}{\|X_j\|_F} \le \frac{\|\tilde{X} - X\|_F / \|\tilde{X}_j\|_F}{1 - \|\tilde{X} - X\|_F / \|\tilde{X}_j\|_F} \le \frac{r(\epsilon)/\|\tilde{X}_j\|_F}{1 - r(\epsilon)/\|\tilde{X}_j\|_F} \equiv b_{\mathrm{rel}}(\tilde{X}_j). \tag{4.8}$$

Proof of Theorem 4.1. The proof is completed by the following three steps.

Step 1. Perturbation equation.

Let

$$X = \mathrm{diag}\{X_j\}_{j=1}^p, \qquad \tilde{X} = \mathrm{diag}\{\tilde{X}_j\}_{j=1}^p,$$

$$\Delta X = \mathrm{diag}\{\Delta X_j\}_{j=1}^p \quad \text{with} \quad \Delta X_j = \tilde{X}_j - X_j, \quad j = 1, \ldots, p,$$

$$A = \mathrm{cyc}\{A_j\}_{j=1}^p, \qquad G = \mathrm{diag}\{G_j\}_{j=1}^p,$$

$$H = \mathrm{diag}(H_2, \ldots, H_p, H_1), \qquad \widehat{R} = \mathrm{diag}(\widehat{R}_2, \ldots, \widehat{R}_p, \widehat{R}_1).$$

Then (1.1) and (4.1) can be expressed by

$$X = A^H X (I + GX)^{-1} A + H \tag{4.9}$$

and

$$\widehat{R} = \tilde{X} - A^H\tilde{X}(I+G\tilde{X})^{-1}A - H, \tag{4.10}$$

respectively. By simple matrix operations, we can get from (4.9) and (4.10) the perturbation equation [16, section 3]

$$\begin{aligned}\Delta X - A^H(I+\tilde{X}G)^{-1}\Delta X(I+G\tilde{X})^{-1}A &= \widehat{R}\\ +A^H(I+\tilde{X}G)^{-1}\Delta X(I+G\tilde{X})^{-1}G\Delta X&\left[I+(I+G\tilde{X})^{-1}G\Delta X\right]^{-1}(I+G\tilde{X})^{-1}A,\end{aligned} \tag{4.11}$$

or equivalently,

$$\mathbf{L}(\Delta X_1,\ldots,\Delta X_{p-1},\Delta X_p) = (\widehat{R}_2,\ldots,\widehat{R}_p,\widehat{R}_1) + (f_2(\Delta X_2),\ldots,f_p(\Delta X_p),f_1(\Delta X_1)), \tag{4.12}$$

where $\mathbf{L}$ is the linear operator defined by (4.2), $\widehat{R}_j$ $(j=1,\ldots,p)$ are the residuals defined by (4.1), and the functions $f_j(\Delta X_j)$ $(j=1,\ldots,p)$ are defined by

$$\begin{aligned}&f_j(\Delta X_j)\\ &= A_j^H(I+\tilde{X}_jG_j)^{-1}\Delta X_j(I+G_j\tilde{X}_j)^{-1}G_j\Delta X_j[I+(I+G_j\tilde{X}_j)^{-1}G_j\Delta X_j]^{-1}\\ &\quad\times(I+G_j\tilde{X}_j)^{-1}A_j.\end{aligned} \tag{4.13}$$

Since the matrix set $\{\tilde{\Phi}_j\}_{j=1}^p$ is pd-stable, the operator $\mathbf{L}$ is invertible. Consequently, the perturbation equation (4.12) can be expressed by

$$\begin{aligned}&(\Delta X_1,\ldots,\Delta X_{p-1},\Delta X_p)\\ &\quad= \mathbf{L}^{-1}[(\widehat{R}_2,\ldots,\widehat{R}_p,\widehat{R}_1) + (f_2(\Delta X_2),\ldots,f_p(\Delta X_p),f_1(\Delta X_1))].\end{aligned} \tag{4.14}$$

Define the function $g(\Delta X_1,\ldots,\Delta X_{p-1},\Delta X_p)$ on $\mathcal{H}_n^p$ by

$$\begin{aligned}&g(\Delta X_1,\ldots,\Delta X_{p-1},\Delta X_p)\\ &\quad= \mathbf{L}^{-1}[(\widehat{R}_2,\ldots,\widehat{R}_p,\widehat{R}_1) + (f_2(\Delta X_2),\ldots,f_p(\Delta X_p),f_1(\Delta X_1))].\end{aligned} \tag{4.15}$$

Obviously, $g(\ )$ can be regarded as a continuous mapping $\mathcal{M}:\mathcal{H}_n^p\to\mathcal{H}_n^p$, and the set of solutions to (4.14) is just the set of fixed points of the mapping $\mathcal{M}$.

Step 2. Estimates of some fixed points of $\mathcal{M}$.

From the definition (4.15) we get

$$\|g(\Delta X_1,\ldots,\Delta X_p)\|_F \le \epsilon + \frac{\|(f_2(\Delta X_2),\ldots,f_p(\Delta X_p),f_1(\Delta X_1))\|_F}{l}, \tag{4.16}$$

where ϵ and l are defined by (4.5). Moreover, from (4.13) we get

$$\|f_j(\Delta X_j)\|_F \le \frac{\phi_j^2\gamma_j\|\Delta X_j\|_F^2}{1-\gamma_j\|\Delta X_j\|_F^2} \le \frac{\phi^2\gamma\|\Delta X_j\|_F^2}{1-\gamma\|(\Delta X_1,\ldots,\Delta X_p)\|_F}, \quad j=1,\ldots,p, \tag{4.17}$$

where $\phi_j, \phi, \gamma_j, \gamma$ are defined by (4.4). Here we assume that the set $\{\Delta X_j\}_{j=1}^p$ satisfies

$$1 - \gamma\|(\Delta X_1, \ldots, \Delta X_p)\|_F > 0. \tag{4.18}$$

Combining (4.16) and (4.17) gives

$$\|g(\Delta X_1, \ldots, \Delta X_p)\|_F \le \epsilon + \frac{\phi^2\gamma\|(\Delta X_1, \ldots, \Delta X_p)\|_F^2}{l(1 - \gamma\|(\Delta X_1, \ldots, \Delta X_p)\|_F)}. \tag{4.19}$$

By using the technique described by [16, section 4] we can prove that if ϵ satisfies the condition (4.6), then the mapping $\mathcal{M}$ has a fixed point $(\Delta X_1^*, \ldots, \Delta X_p^*)$ in the set

$$\mathcal{S}_{r(\epsilon)} = \{(\Delta X_1, \ldots, \Delta X_p) \in \mathcal{H}_n^p : \ \|(\Delta X_1, \ldots, \Delta X_p)\|_F \le r(\epsilon)\}, \tag{4.20}$$

where $r(\epsilon)$ is defined by (4.7).

Note that the condition (4.6) implies that for any $(\Delta X_1, \ldots, \Delta X_p) \in \mathcal{S}_{r(\epsilon)}$ the inequality (4.18) holds. In fact, if $(\Delta X_1, \ldots, \Delta X_p) \in \mathcal{S}_{r(\epsilon)}$, then we have

$$\begin{aligned}
\gamma\|(\Delta X_1, \ldots, \Delta X_p)\|_F &\le \gamma r(\epsilon) \quad \text{(by (4.20))} \\
&\le \frac{2\gamma\epsilon}{1 + \gamma\epsilon} \quad \text{(by (4.7))} \\
&< \frac{l}{\phi^2 + \phi\sqrt{\phi^2 + l} + l} \quad \text{(by (4.6))} \\
&\le 1.
\end{aligned}$$

Step 3. On the matrix set $\{\tilde{X}_j - \Delta X_j^*\}_{j=1}^p$.

Let

$$\Delta X^* = \mathrm{diag}(\Delta X_1^*, \ldots, \Delta X_p^*)$$

and

$$Y = \tilde{X} - \Delta X^* = \mathrm{diag}(Y_1, \ldots, Y_p).$$

Then from Step 1 we see that Y is a Hermitian solution to the DARE (4.9); i.e., Y satisfies

$$Y = A^H Y(I + GY)^{-1}A + H, \tag{4.21}$$

or equivalently,

$$\begin{aligned}
&Y - A^H(I + YG)^{-1}Y(I + GY)^{-1}A \\
&\quad = H + A^H(I + YG)^{-1}YGY(I + GY)^{-1}A.
\end{aligned} \tag{4.22}$$

Observe the following two facts:

1. The matrix on the right-hand side of (4.22) is Hermitian p.s.d.
2. The matrix $(I + GY)^{-1}A$ can be written as

$$(I + GY)^{-1}A = [I - (I + G\tilde{X})^{-1}G\Delta X^*](I + G\tilde{X})^{-1}A,$$

or equivalently,

$$\text{cyc}\left\{(I+G_jY_j)^{-1}A_j\right\}_{j=1}^p = \text{cyc}\left\{\left[I-(I+G_j\tilde{X}_j)^{-1}G_j\Delta X_j^*\right]^{-1}\tilde{\Phi}_j\right\}_{j=1}^p, \tag{4.23}$$

where the matrices $\tilde{\Phi}_j$ are defined by (4.3), and by the hypotheses the matrix set $\{\tilde{\Phi}_j\}_{j=1}^p$ is pd-stable. Moreover, for $j=1,\ldots,p$ we have

$$\begin{aligned}
&\left\|(I+G_j\tilde{X}_j)^{-1}G_j\Delta X_j^*\right\|_2\\
&\le \left\|(I+G_j\tilde{X}_j)^{-1}G_j\right\|_2\|\Delta X_j^*\|_2\\
&\le \gamma_j r(\epsilon) \quad \text{(by (4.4) and (4.20))}\\
&\le \frac{2\gamma l\epsilon}{(1+\gamma\epsilon)l+\sqrt{(1+\gamma\epsilon)^2l^2-4(\phi^2+l)\gamma l\epsilon}} \quad \text{(by (4.4) and (4.7))}\\
&\le \frac{2\gamma\epsilon}{1+\gamma\epsilon}\\
&< \frac{l}{\phi^2+\phi\sqrt{\phi^2+l}+l} \quad \text{(by (4.6))}.
\end{aligned}$$

Consequently, by Lemma 2.4, the matrix set $\{[I-(I+G_j\tilde{X}_j)^{-1}G_j\Delta X_j^*]^{-1}\tilde{\Phi}_j\}_{j=1}^p$ is pd-stable. By (4.23), the matrix set $\left\{(I+G_jY_j)^{-1}A_j\right\}_{j=1}^p$ is pd-stable. Further, by Lemma 2.2, the matrix

$$\text{cyc}\left\{(I+G_jY_j)^{-1}A_j\right\}_{j=1}^p = (I+GY)^{-1}A$$

is d-stable.

Hence, the Hermitian matrix $Y=\text{diag}(Y_1,\ldots,Y_p)$, as a solution to (4.22), is positive semidefinite [3, Proposition 2.1]; and so the matrix Y, as a Hermitian solution to the DARE (4.21), is positive semidefinite and stabilizing. By the uniqueness of the stabilizing solution to the DARE (4.21) [8, Proposition 1], we have $Y=X=\text{diag}\{X_j\}_{j=1}^p$, the unique Hermitian p.s.d. stabilizing solution to the DARE (4.9). Thus, the matrix set $\{Y_j\}_{j=1}^p$ is just the unique Hermitian p.s.d. stabilizing solution set to the P-DARE (1.1).

Overall, we have proved the estimate

$$\|(\tilde{X}_1-X_1,\ldots,\tilde{X}_p-X_p)\|_F = \|(\Delta X_1^*,\ldots,\Delta X_p^*)\|_F \le r(\epsilon). \qquad \square$$

Note that the function $r(\epsilon)$ defined by (4.7) has the Taylor expansion at $\epsilon=0$:

$$r(\epsilon)=\epsilon+\frac{\gamma\phi^2}{l}\epsilon^2+O(\epsilon^3) \quad \text{as } \epsilon\to 0.$$

Consequently, for sufficiently small $\|\mathbf{L}^{-1}(\widehat{R}_1,\ldots,\widehat{R}_p)\|_F$, we have the first order estimate

$$\|(\tilde{X}_1-X_1,\ldots,\tilde{X}_p-X_p)\|_F \lesssim \epsilon = \left\|\mathbf{L}^{-1}(\widehat{R}_1,\ldots,\widehat{R}_p)\right\|_F.$$

5. Numerical results. We now use a simple numerical example to illustrate our results of sections 3 and 4. All computations were performed using MATLAB, version 6.1. The relative machine precision is 2.22×10^{-16}.

Example 5.1. Consider the P-DARE (1.1) with $n = 2,\ p = 3$, and

$$\begin{aligned} &A_1 = \begin{pmatrix} 0 & 0 \\ 10^m & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \\ &B_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad B_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \\ &C_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad C_2 = (1,\ 0), \quad C_3 = (0,\ 0), \\ &R_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad R_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad R_3 = 1. \end{aligned} \tag{5.1}$$

By $G_j = B_j R_j^{-1} B_j^T$ and $H_j = C_j^T C_j$ $(j = 1, 2, 3)$, we have

$$\begin{aligned} &G_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad G_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad G_3 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \\ &H_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad H_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad H_3 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}. \end{aligned} \tag{5.2}$$

Thus, the corresponding P-DARE (1.1) can be written

$$\begin{cases} X_3 = A_1^T (I + G_1 X_1)^{-1} A_1 + H_1, \\ X_1 = A_2^T (I + G_2 X_2)^{-1} A_2 + H_2, \\ X_2 = A_3^T (I + G_3 X_3)^{-1} A_3 + H_3, \end{cases} \tag{5.3}$$

where A_j and G_j, H_j are the matrices of (5.1) and (5.2), respectively.

By (5.1), (2.14), and (2.15), we get the matrices $\mathcal{A}_j$, $\mathcal{B}_j$, and $\mathcal{C}_j$ $(j = 1, 2, 3)$ with

$$\mathcal{A}_1 = A_3 A_2 A_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},$$

$$\mathcal{B}_1 = (A_3 A_2 B_1,\ A_3 B_2,\ B_3) = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

$$\mathcal{C}_1 = \left(C_1^T,\ A_1^T C_2^T,\ A_1^T A_2^T C_3^T\right)^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}^T,$$

$$\mathcal{A}_2 = A_2 A_1 A_3 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},$$

$$\mathcal{B}_2 = (A_2 A_1 B_3,\ A_2 B_1,\ B_2) = \begin{pmatrix} 10^m & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\mathcal{C}_2 = \left(C_3^T,\ A_3^T C_1^T,\ A_3^T A_1^T C_2^T\right)^T = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}^T,$$

$$\mathcal{A}_3 = A_1 A_3 A_2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},$$

$$\mathcal{B}_3 = (A_1 A_3 B_2,\ A_1 B_3,\ B_1) = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 10^m & 0 & 1 \end{pmatrix},$$

$$\mathcal{C}_3 = \left(C_2^T,\ A_2^T C_3^T,\ A_2^T A_3^T C_1^T\right)^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}^T.$$

TABLE 5.1
Estimates of relative backward errors ($k = 12$).

m	l^*	u^*	$c_{\rm rel}(X_1, X_2, X_3)$
0	8.8×10^{-13}	8.8×10^{-13}	2.7
1	5.7×10^{-11}	5.7×10^{-11}	7.1×10
2	5.7×10^{-9}	5.7×10^{-9}	7.1×10^3
3	3.5×10^{-7}	7.8×10^{-7}	7.1×10^5
4	$*$	$*$	7.1×10^7

It can be verified that the matrix pairs $(\mathcal{A}_j, \mathcal{B}_j)$ are d-stabilizable, and $(\mathcal{A}_j, \mathcal{C}_j)$ are d-detectable for $j = 1, 2, 3$; i.e., the matrix pair sets $\{(A_j, B_j)\}_{j=1}^3$ and $\{(A_j, C_j)\}_{j=1}^3$ are pd-stabilizable and pd-detectable, respectively. By Theorem 2.5, the P-DARE (5.3) has a unique symmetric p.s.d. stabilizing solution set $\{X_j\}_{j=1}^3$. It is easy to verify that the set $\{X_j\}_{j=1}^3$ with

$$X_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad X_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad X_3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

is the unique symmetric p.s.d. stabilizing solution set, which is independent of the values of m.

Let the approximate symmetric p.s.d. solution sets $\{\tilde{X}_j\}_{j=1}^3$ be given by

(5.4)
$$\tilde{X}_1 = X_1 + \begin{pmatrix} -0.3 & -0.2 \\ -0.2 & 0.8 \end{pmatrix} \times 10^{-k}, \quad \tilde{X}_2 = X_2 + \begin{pmatrix} 0.1 & 0.1 \\ 0.1 & -0.2 \end{pmatrix} \times 10^{-k},$$

$$\tilde{X}_3 = X_3 + \begin{pmatrix} -0.2 & 0.3 \\ 0.3 & 0.6 \end{pmatrix} \times 10^{-k}, \quad k = 0, 1, 2, \ldots.$$

We now are going to give estimates of backward errors and residual bounds for the approximate symmetric p.s.d. solution sets.

Estimates of backward errors. Some numerical results on backward errors of the approximate solution sets are listed in Tables 5.1 and 5.2, where the bounds l^* and u^* are computed by (3.17)–(3.18), and the values of the relative condition number $c_{\rm rel}(X_1, X_2, X_3)$ listed in Table 5.1 are computed by [12, (4.24)] with

$$\xi_j = \|X_j\|_F, \quad \alpha_j = \|A_j\|_F, \quad \gamma_j = \|G_j\|_F, \quad \eta_j = \|H_j\|_F, \quad j = 1, 2, 3.$$

The cases when the condition (3.15) of Theorem 3.1 is violated are denoted by asterisks in Tables 5.1 and 5.2.

From the results listed in Table 5.1 we see that the relative backward error increases as the relative conditioning of the P-DARE deteriorates.

The results listed in Tables 5.1 and 5.2 show that the relative backward errors are very small ($\eta(\{\tilde{X}_j\}_{j=1}^3) \lesssim 5.7 \times 10^{-9}$) in the cases of $k = 12$ and $m \le 2$, and in the cases of $m = 1$ and $k \ge 10$; this means that in such cases each approximate symmetric p.s.d. solution set $\{\tilde{X}_j\}_{j=1}^3$ is an exact symmetric p.s.d. solution set to a slightly perturbed P-DARE.

From the results listed in Table 5.2 we see that the relative backward error decreases as the error $\|\tilde{X} - X\|_F = \sqrt{\sum_{j=1}^3 \|\tilde{X}_j - X_j\|_F^2}$ decreases.

TABLE 5.2
Estimates of relative backward errors ($m = 1$, $c_{\rm rel}(X_1, X_2, X_3) \approx 71$).

k	l^*	u^*	$\|\tilde{X} - X\|_F$
2	$*$	$*$	1.2×10^{-2}
4	3.4×10^{-3}	7.9×10^{-3}	1.2×10^{-4}
6	5.7×10^{-5}	5.7×10^{-5}	1.2×10^{-6}
8	5.7×10^{-7}	5.7×10^{-7}	1.2×10^{-8}
10	5.7×10^{-9}	5.7×10^{-9}	1.2×10^{-10}
12	5.7×10^{-11}	5.7×10^{-11}	1.2×10^{-12}

Computed results for this example show that

$$l_j \approx u_j \approx \rho_j, \quad j = 1, 2, 3,$$

which mean that the linear estimates (3.19) are relatively sharp, while the nonlinear estimates (3.16) do not even exist in some cases. However, it is worth pointing out that the nonlinear estimates (3.16) guarantee the existence of the solution to the optimization problem (3.5), while the linear estimates (3.19) would formally give approximate bounds which might not correspond to any solution to the problem (3.5).

Residual bounds. Here we only present a few results in the case of $m = 0$. In such a case, the relative condition number $c_{\rm rel}(X_1, X_2, X_3) \approx 2.7$. Taking $k = 4$ in (5.4) we obtain an approximate symmetric p.s.d. solution set $\{\tilde{X}_j\}_{j=1}^3$, among which each $\tilde{X}_j$ approximates X_j ($1 \le j \le 3$) up to 5 significant figures.

A computation by (4.7) gives

$$\frac{r(\epsilon)}{\|\tilde{X}_1\|_F} \approx 1.5 \times 10^{-4}, \quad \frac{r(\epsilon)}{\|\tilde{X}_2\|_F} \approx 1.5 \times 10^{-4}, \quad \frac{r(\epsilon)}{\|\tilde{X}_3\|_F} \approx 1.1 \times 10^{-4}.$$

Combining the estimates with (4.8) we get relative error bounds for $\tilde{X}_j$ ($j = 1, 2, 3$):

$$b_{\rm rel}(\tilde{X}_1) \approx 1.5 \times 10^{-4}, \quad b_{\rm rel}(\tilde{X}_2) \approx 1.5 \times 10^{-4}, \quad b_{\rm rel}(\tilde{X}_3) \approx 1.1 \times 10^{-4}. \tag{5.5}$$

From (5.5) we see that the approximate symmetric p.s.d. solution set $\{\tilde{X}_j\}_{j=1}^3$ has at least 4 correct digits.

Note that by Theorem 4.1 the estimate (4.7) can only be applied to the case where the condition (4.6) is satisfied; i.e.,

$$\delta(\epsilon) \equiv \frac{l}{\gamma(2\phi^2 + 2\phi\sqrt{\phi^2 + l} + l)} - \epsilon > 0.$$

The results listed in Table 5.3 show the scope of application of the estimate (4.7) for this example.

Example 5.2 (see [12, Example 5.1]). Consider the P-DARE (1.1) with $n = 3, p = 3$, and

$$A_j = V_j^T A_j^{(0)} V_j, \quad G_j = V_j^T G_j^{(0)} V_j, \quad H_j = V_j^T H_j^{(0)} V_j, \quad j = 1, 2, 3,$$

TABLE 5.3

m	$c_{\rm rel}(X_1, X_2, X_3)$	$\delta(\epsilon)$
0	2.7	$\delta(\epsilon) > 0$ if and only if $k \geq 1$
1	7.1×10	$\delta(\epsilon) > 0$ if and only if $k \geq 11$
2	7.1×10^3	$\delta(\epsilon) > 0$ if and only if $k \geq 13$
3	7.1×10^5	$\delta(\epsilon) > 0$ if and only if $k \geq 21$

TABLE 5.4
Estimates of relative backward errors.

m	l^*	u^*
0	2.2×10^{-15}	2.2×10^{-15}
1	1.9×10^{-15}	1.9×10^{-15}
2	3.8×10^{-15}	3.8×10^{-15}
3	1.9×10^{-14}	1.9×10^{-14}
4	4.0×10^{-13}	4.0×10^{-13}
5	5.0×10^{-12}	5.0×10^{-12}
6	5.0×10^{-11}	5.0×10^{-11}

where

$$A_1^{(0)} = \mathrm{diag}(0, 10^{-m}, 1), \quad A_2^{(0)} = \mathrm{diag}(10^{-9}, 10^{-m}, 1 + 10^{-3}),$$

$$A_3^{(0)} = \mathrm{diag}(10^{-3}, 10^{-m+1}, 0.5),$$

$$G_j^{(0)} = \mathrm{diag}\left(\frac{1}{j}10^{-m}, \frac{1}{j}10^{-m}, j \times 10^{-m}\right), \quad H_j^{(0)} = \mathrm{diag}\left(\frac{1}{j}10^{m}, j, j \times 10^{-m}\right),$$

$j = 1, 2, 3,$

and

$$V_1 = I - 2v_1v_1^T \quad \text{with} \quad v_1 = \frac{1}{\sqrt{3}}(1, 1, 1)^T,$$

$$V_2 = I - 2v_2v_2^T \quad \text{with} \quad v_2 = \frac{1}{\sqrt{6}}(1, 1, 2)^T,$$

$$V_3 = I - 2v_3v_3^T \quad \text{with} \quad v_3 = \frac{1}{\sqrt{11}}(-1, 1, 3)^T.$$

By applying the file "dare" of Control System Toolbox, we get computed symmetric p.s.d. solution sets $\{\tilde{X}_1, \tilde{X}_2, \tilde{X}_3\}$ to the P-DARE (1.1). Some numerical results on backward errors and residual bounds for the computed solution sets are listed in Tables 5.4 and 5.5, respectively, where the relative error bounds $b_{\rm rel}(\tilde{X}_j)$ for $\tilde{X}_j$ $(j = 1, 2, 3)$ are defined by (4.8).

The results listed in Table 5.4 show that each computed symmetric p.s.d. solution set $\{\tilde{X}_1, \tilde{X}_2, \tilde{X}_3\}$ by applying the file "dare" of Control System Toolbox is the exact symmetric p.s.d. solution set to a slightly perturbed P-DARE; in other words, the computation has proceeded stably.

From the results listed in Table 5.5 we see that the computed symmetric p.s.d. solution sets $\{\tilde{X}_1, \tilde{X}_2, \tilde{X}_3\}$ have high relative precision when m is a small natural number; e.g., in the case $m = 3$, each computed $\tilde{X}_j$ has at least 14 correct digits.

TABLE 5.5
Residual bounds.

m	ϵ	$r(\epsilon)$	$b_{\rm rel}(\tilde{X}_1)$	$b_{\rm rel}(\tilde{X}_2)$	$b_{\rm rel}(\tilde{X}_3)$
0	8.3×10^{-13}	8.3×10^{-13}	2.1×10^{-13}	7.5×10^{-15}	3.2×10^{-13}
1	1.3×10^{-14}	1.3×10^{-14}	2.3×10^{-15}	2.3×10^{-15}	1.3×10^{-15}
2	3.3×10^{-13}	3.3×10^{-13}	6.2×10^{-15}	9.6×10^{-15}	3.2×10^{-15}
3	2.0×10^{-11}	2.0×10^{-11}	3.8×10^{-14}	6.0×10^{-14}	2.0×10^{-14}
4	4.2×10^{-9}	4.2×10^{-9}	8.0×10^{-13}	1.2×10^{-12}	4.1×10^{-13}
5	5.4×10^{-7}	5.4×10^{-7}	1.0×10^{-11}	1.6×10^{-11}	5.3×10^{-12}
6	5.3×10^{-5}	5.3×10^{-5}	1.0×10^{-10}	1.6×10^{-10}	5.3×10^{-11}

Acknowledgment. I would like to thank Wen-Wei Lin, who gave me the computed symmetric p.s.d. solution sets $\{\tilde{X}_1, \tilde{X}_2, \tilde{X}_3\}$ to the P-DARE (1.1) of Example 5.2 by applying the file "dare" of Control System Toolbox. I also thank the referees for helpful comments and suggestions.

REFERENCES

[1] S. BITTANTI, P. COLANERI, AND G. DE NICOLAO, *The difference periodic Riccati equation for the periodic prediction problem*, IEEE Trans. Automat. Control, 33 (1988), pp. 706–712.
[2] S. BITTANTI, P. COLANERI, AND G. DE NICOLAO, *The periodic Riccati equation*, in The Riccati Equations, S. Bittanti, A. Laub, and J. Willems, eds., Springer-Verlag, Berlin, 1991.
[3] P. M. GAHINET, A. J. LAUB, C. S. KENNEY, AND G. A. HEWER, *Sensitivity of the stable discrete-time Lyapunov equation*, IEEE Trans. Automat. Control, 35 (1990), pp. 1209–1217.
[4] A. R. GHAVIMI AND A. J. LAUB, *Backward error, sensitivity, and refinement of computed solutions of algebraic Riccati equations*, Numer. Linear Algebra Appl., 2 (1995), pp. 29–49.
[5] A. GRAHAM, *Kronecker Products and Matrix Calculus: With Applications*, John Wiley, New York, 1981.
[6] N. J. HIGHAM, *Perturbation theory and backward error for $AX - XB = C$*, BIT, 33 (1993), pp. 124–136.
[7] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
[8] V. IONESCU AND M. WEISS, *On computing the stabilizing solution of the discrete-time Riccati equation*, Linear Algebra Appl., 174 (1992), pp. 229–238.
[9] B. KÅGSTRÖM, *A perturbation analysis of the generalized Sylvester equation $(AR - LB, DR - LE) = (C, F)$*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1045–1060.
[10] M. M. KONSTANTINOV AND P. HR. PETKOV, *Note on "Perturbation theory for algebraic Riccati equations,"* SIAM J. Matrix Anal. Appl., 21 (1999), p. 327.
[11] M. M. KONSTANTINOV, P. HR. PETKOV, V. MEHRMANN, AND D. GU, *Additive matrix operators*, in Proceedings of the 30th Spring Conference of Bulgarian Mathematicians, Borovets (Bulgaria), 2001, pp. 169–175.
[12] W.-W. LIN AND J.-G. SUN, *Perturbation analysis of the periodic discrete-time algebraic Riccati equation*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 411–438.
[13] G. W. STEWART AND J.-G. SUN, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.
[14] J.-G. SUN, *Residual bounds of approximate solutions of the algebraic Riccati equation*, Numer. Math., 76 (1997), pp. 249–263.
[15] J.-G. SUN, *Backward error for the discrete-time algebraic Riccati equation*, Linear Algebra Appl., 259 (1997), pp. 183–208.
[16] J.-G. SUN, *Residual bounds of approximate solutions of the discrete-time algebraic Riccati equation*, Numer. Math., 78 (1998), pp. 463–478.
[17] J.-G. SUN, *Stability and Accuracy: Perturbation Analysis of Algebraic Eigenproblems*, Report UMINF 98.07 (revised), ISSN-0348-0542, Department of Computing Science, Umeå University, 2002.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 1, pp. 20–41

A PRECONDITIONER FOR GENERALIZED SADDLE POINT PROBLEMS*

MICHELE BENZI† AND GENE H. GOLUB‡

Abstract. In this paper we consider the solution of linear systems of saddle point type by preconditioned Krylov subspace methods. A preconditioning strategy based on the symmetric/skew-symmetric splitting of the coefficient matrix is proposed, and some useful properties of the preconditioned matrix are established. The potential of this approach is illustrated by numerical experiments with matrices from various application areas.

Key words. saddle point problems, matrix splittings, iterative methods, preconditioning

AMS subject classifications. Primary, 65F10, 65N22, 65F50; Secondary, 15A06

DOI. 10.1137/S0895479802417106

1. Introduction.

We consider the solution of systems of linear equations with the following block 2×2 structure:

$$\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times n}$, $C \in \mathbb{R}^{m\times m}$, $f \in \mathbb{R}^n$, $g \in \mathbb{R}^m$, and $m \le n$. We further assume that matrices A, B, and C are large and sparse. Systems of the form (1.1) arise in a variety of scientific and engineering applications, including computational fluid dynamics [1, 20, 22, 24, 27, 44], mixed finite element approximation of elliptic PDEs [12, 48, 60], optimization [5, 25, 26, 32, 39, 43], optimal control [9, 35], weighted and equality constrained least squares estimation [10], structural analysis [56], electrical networks [56], inversion of geophysical data [34], computer graphics [42], and others.

An important special case of (1.1) is when A is symmetric positive semidefinite, $C = O$, $\mathrm{rank}(B) = m$, and $\mathcal{N}(A) \cap \mathcal{N}(B) = \{0\}$. In this case (1.1) corresponds to a saddle point problem, and it has a unique solution.

In this paper we consider *generalized* saddle point problems, i.e., systems of the form (1.1) satisfying all of the following assumptions:

- A has positive semidefinite symmetric part $H = \frac{1}{2}(A + A^T)$;
- $\mathrm{rank}(B) = m$;
- $\mathcal{N}(H) \cap \mathcal{N}(B) = \{0\}$;
- C is symmetric positive semidefinite.

As shown below (Lemma 1.1), these assumptions guarantee existence and uniqueness of the solution. Although very often A is symmetric positive definite, we are especially interested in cases where A is either symmetric and singular (i.e., only positive semidefinite), or nonsymmetric with positive definite symmetric part H (i.e., A is *positive real*). The latter situation arises when the steady-state Navier–Stokes

*Received by the editors October 30, 2002; accepted for publication (in revised form) by A. J. Wathen October 27, 2003; published electronically August 6, 2004.
http://www.siam.org/journals/simax/26-1/41710.html

†Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322 (benzi@mathcs.emory.edu). The work of this author was supported in part by National Science Foundation grant DMS-0207599.

‡Scientific Computing and Computational Mathematics Program, Stanford University, Stanford, CA 94305-9025 (golub@sccm.stanford.edu). The work of this author was supported in part by Department of Energy grant DOE-FC02-01ER4177.

equations are linearized by a Picard iteration, leading to the *Oseen equations*; see [20, 22]. In this case, the A block corresponds to an appropriate discretization of a convection-diffusion operator.

A number of solution methods have been proposed in the literature. Besides specialized sparse direct solvers [16, 17] we mention, among others, Uzawa-type schemes [11, 21, 24, 27, 62], block and approximate Schur complement preconditioners [4, 15, 20, 22, 41, 45, 46, 48, 51], splitting methods [18, 30, 31, 49, 57], indefinite preconditioning [23, 35, 39, 43, 48], iterative projection methods [5], iterative null space methods [1, 32, 54], and preconditioning methods based on approximate factorization of the coefficient matrix [25, 50]. Several of these algorithms are based on some form of reduction to a smaller system, for example, by projecting the problem onto the null space of B, while others work with the original (augmented) matrix in (1.1). The method studied in this paper falls in the second category.

When A is symmetric positive (semi-)definite, the coefficient matrix in (1.1) is symmetric indefinite, and indefinite solvers can be used to solve problem (1.1). Alternatively, one can solve instead of (1.1) the equivalent nonsymmetric system

$$\begin{bmatrix} A & B^T \\ -B & C \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ -g \end{bmatrix}, \quad \text{or} \quad \mathcal{A}\mathbf{x} = \mathbf{b}, \tag{1.2}$$

where $\mathcal{A}$ is the coefficient matrix in (1.2), $\mathbf{x} = [u^T, p^T]^T$ and $\mathbf{b} = [f^T, -g^T]^T$. The nonsymmetric formulation is especially natural when A is nonsymmetric, but positive real. Whether A is symmetric or not, the nonsymmetric matrix $\mathcal{A}$ has certain desirable properties, which are summarized in the following result.

LEMMA 1.1. *Let $\mathcal{A} \in \mathbb{R}^{(n+m)\times(n+m)}$ be the coefficient matrix in* (1.2). *Assume $H = \frac{1}{2}(A + A^T)$ is positive semidefinite, B has full rank, $C = C^T$ is positive semidefinite, and $\mathcal{N}(H) \cap \mathcal{N}(B) = \{0\}$. Let $\sigma(\mathcal{A})$ denote the spectrum of $\mathcal{A}$. Then*

(i) *$\mathcal{A}$ is nonsingular.*

(ii) *$\mathcal{A}$ is* semipositive real*: $\langle \mathcal{A}\mathbf{v}, \mathbf{v}\rangle = \mathbf{v}^T\mathcal{A}\mathbf{v} \geq 0$ for all $\mathbf{v} \in \mathbb{R}^{n+m}$.*

(iii) *$\mathcal{A}$ is* positive semistable*, that is, the eigenvalues of $\mathcal{A}$ have nonnegative real part: $\Re(\lambda) \geq 0$ for all $\lambda \in \sigma(\mathcal{A})$.*

(iv) *If, in addition, $H = \frac{1}{2}(A + A^T)$ is positive definite, then $\mathcal{A}$ is* positive stable*: $\Re(\lambda) > 0$ for all $\lambda \in \sigma(\mathcal{A})$.*

Proof. To prove (i), let $\mathbf{x} = \left[\begin{smallmatrix} u \\ p \end{smallmatrix}\right]$ be such that $\mathcal{A}\mathbf{x} = \mathbf{0}$. Then

$$Au + B^T p = 0 \quad \text{and} \quad -Bu + Cp = 0. \tag{1.3}$$

Now, from $\mathcal{A}\mathbf{x} = \mathbf{0}$ we get $\mathbf{x}^T\mathcal{A}\mathbf{x} = u^TAu + p^TCp = 0$, and therefore it must be $u^TAu = 0$ and $p^TCp = 0$, since both of these quantities are nonnegative. But $u^TAu = u^THu = 0$, which implies $u \in \mathcal{N}(H)$ since H is symmetric positive semidefinite (see [36, p. 400]). Similarly, $p^TCp = 0$ with C symmetric positive semidefinite implies $Cp = 0$ and therefore (using the second of (1.3)) $Bu = 0$. Therefore $u = 0$ since $u \in \mathcal{N}(H) \cap \mathcal{N}(B) = \{0\}$. But if $u = 0$ then from the first of (1.3) we obtain $B^Tp = 0$ and therefore $p = 0$ since B has full column rank. Therefore the only solution to $\mathcal{A}\mathbf{x} = \mathbf{0}$ is the trivial solution, and $\mathcal{A}$ is nonsingular.

To prove (ii) we observe that for any $\mathbf{v} \in \mathbb{R}^{n+m}$ we have $\mathbf{v}^T\mathcal{A}\mathbf{v} = \mathbf{v}^T\mathcal{H}\mathbf{v}$, where

$$\mathcal{H} = \tfrac{1}{2}(\mathcal{A} + \mathcal{A}^T) = \begin{bmatrix} H & O \\ O & C \end{bmatrix}$$

is the symmetric part of $\mathcal{A}$. Clearly $\mathcal{H}$ is positive semidefinite, hence $\mathbf{v}^T\mathcal{A}\mathbf{v} \geq 0$.

To prove (iii), let $(\lambda, \mathbf{v})$ be an eigenpair of $\mathcal{A}$, with $||\mathbf{v}||_2 = 1$. Then $\mathbf{v}^*\mathcal{A}\mathbf{v} = \lambda$ and $(\mathbf{v}^*\mathcal{A}\mathbf{v})^* = \mathbf{v}^*\mathcal{A}^T\mathbf{v} = \bar{\lambda}$. Therefore $\frac{1}{2}\mathbf{v}^*(\mathcal{A} + \mathcal{A}^T)\mathbf{v} = \frac{\lambda+\bar{\lambda}}{2} = \Re(\lambda)$. To conclude the proof, observe that

$$\mathbf{v}^*(\mathcal{A} + \mathcal{A}^T)\mathbf{v} = \Re(\mathbf{v})^T(\mathcal{A} + \mathcal{A}^T)\Re(\mathbf{v}) + \Im(\mathbf{v})^T(\mathcal{A} + \mathcal{A}^T)\Im(\mathbf{v}),$$

a real nonnegative quantity.

To prove (iv), assume $(\lambda, \mathbf{v})$ is an eigenpair of $\mathcal{A}$ with $\mathbf{v} = \left[\begin{smallmatrix} u \\ p \end{smallmatrix}\right]$. Then

$$\Re(\lambda) = u^*Hu + p^*Cp = \Re(u)^TH\Re(u) + \Im(u)^TH\Im(u) + \Re(p)^TC\Re(p) + \Im(p)^TC\Im(p).$$

This quantity is nonnegative, and it can be zero only if $u = 0$ (since H is assumed to be positive definite) and $Cp = 0$. But if $u = 0$ then from the first of (1.3) we get $B^Tp = 0$, hence $p = 0$ since B has full column rank. Hence $\mathbf{v} = \mathbf{0}$, a contradiction. □

Thus, by changing the sign of the last m equations in (1.1) we may lose symmetry (when A is symmetric), but we gain positive (semi-)definiteness. This can be advantageous when using certain Krylov subspace methods, like restarted GMRES; see [19, 53].

In this paper we propose a new approach for preconditioning generalized saddle point problems based on an alternating symmetric/skew-symmetric splitting [2] applied to (1.2). This approach is very general in that it does not require the submatrix A to be nonsingular or symmetric; hence, it is applicable to a broad class of problems. The splitting method is described in section 2, and some of its convergence properties are studied in section 3. The use of the splitting as a preconditioner for Krylov subspace methods is considered in section 4. Numerical experiments are presented in section 5. Finally, in section 6 we draw our conclusions.

2. The alternating splitting iteration. In [2], the following stationary iterative methods for solving positive real linear systems $\mathcal{A}\mathbf{x} = \mathbf{b}$ was proposed. Write $\mathcal{A} = \mathcal{H} + \mathcal{S}$, where

$$\mathcal{H} = \tfrac{1}{2}(\mathcal{A} + \mathcal{A}^T), \quad \mathcal{S} = \tfrac{1}{2}(\mathcal{A} - \mathcal{A}^T)$$

are the symmetric and skew-symmetric part of $\mathcal{A}$, respectively. Let $\alpha > 0$ be a parameter. Similar in spirit to the classical alternating direction implicit (ADI) method [58], consider the following two splittings of $\mathcal{A}$:

$$\mathcal{A} = (\mathcal{H} + \alpha\mathcal{I}) - (\alpha\mathcal{I} - \mathcal{S})$$

and

$$\mathcal{A} = (\mathcal{S} + \alpha\mathcal{I}) - (\alpha\mathcal{I} - \mathcal{H}).$$

Here $\mathcal{I}$ denotes the identity matrix. The algorithm is obtained by alternating between these two splittings (see [7] for a general study of alternating iterations). Given an initial guess $\mathbf{x}^0$, the symmetric/skew-symmetric iteration computes a sequence $\{\mathbf{x}^k\}$ as follows:

$$\begin{cases} (\mathcal{H} + \alpha\mathcal{I})\mathbf{x}^{k+\frac{1}{2}} = (\alpha\mathcal{I} - \mathcal{S})\mathbf{x}^k + \mathbf{b}, \\ (\mathcal{S} + \alpha\mathcal{I})\mathbf{x}^{k+1} = (\alpha\mathcal{I} - \mathcal{H})\mathbf{x}^{k+\frac{1}{2}} + \mathbf{b}. \end{cases} \tag{2.1}$$

It is shown in [2] that if $\mathcal{H}$ is positive definite, the stationary iteration (2.1) converges for all $\alpha > 0$ to the solution of $\mathcal{A}\mathbf{x} = \mathbf{b}$.

Let us now consider the application of (2.1) to generalized saddle point problems in the form (1.2). In this case we have

$$\mathcal{H} = \begin{bmatrix} H & O \\ O & C \end{bmatrix} \quad \text{and} \quad \mathcal{S} = \begin{bmatrix} S & B^T \\ -B & O \end{bmatrix},$$

where $S = \frac{1}{2}(A - A^T)$ is the skew-symmetric part of A. Hence, $\mathcal{A}$ is positive real only when submatrices H and C are both symmetric positive definite (SPD), which is almost never the case in practice. Therefore, the convergence theory developed in [2] does not apply, and a more subtle analysis is required. We provide this analysis in the next section.

A few remarks are in order. At each iteration of (2.1), it is required to solve two sparse linear systems with coefficient matrices $\mathcal{H} + \alpha\mathcal{I}$ and $\mathcal{S} + \alpha\mathcal{I}$. Note that under our assumptions, both of these matrices are invertible for all $\alpha > 0$. Clearly, the choice of the solution methods used to perform the two half-steps in (2.1) is highly problem-dependent, and must be done on a case-by-case basis. The alternating algorithm (2.1) is just a general scheme that can incorporate whatever solvers are appropriate for a given problem.

Nevertheless, it is possible to make some general observations. The first half-step of algorithm (2.1) necessitates the solution of two (uncoupled) linear systems of the form

$$\begin{cases} (H + \alpha I_n)u^{k+\frac{1}{2}} = \alpha u^k - Su^k + f - B^T p^k, \\ (C + \alpha I_m)p^{k+\frac{1}{2}} = \alpha p^k - g + Bu^k. \end{cases} \tag{2.2}$$

Both systems in (2.2) are SPD, and any sparse solver for SPD systems can be applied. This could be a sparse Cholesky factorization, or a preconditioned conjugate gradient (PCG) scheme, or some specialized solver. Note that the addition of a positive term α to the main diagonal of H (and C) improves the condition number. This, in turn, tends to improve the rate of convergence of iterative methods applied to (2.2). More precisely, if H is normalized so that its largest eigenvalue is equal to 1, then for the spectral condition number of $H + \alpha I$ we have

$$\kappa(H + \alpha I) = \frac{1 + \alpha}{\lambda_{\min}(H) + \alpha} \leq 1 + \frac{1}{\alpha},$$

independent of the size of the problem. Note that even a fairly small value of α, such as $\alpha = 0.1$, yields a small condition number ($\kappa(H + \alpha I) \leq 11$). Unless α is very small, rapid convergence of the CG method applied to (2.2) can be expected, independent of the number n of unknowns.

The second half-step of algorithm (2.1) is less trivial. It requires the solution of two coupled linear systems of the form

$$\begin{cases} (\alpha I_n + S)u^{k+1} + B^T p^{k+1} = (\alpha I_n - H)u^{k+\frac{1}{2}} + f \equiv f^k, \\ -Bu^{k+1} + \alpha p^{k+1} = (\alpha I_m - C)p^{k+\frac{1}{2}} - g \equiv g^k. \end{cases} \tag{2.3}$$

This system can be solved in several ways. Of course, a sparse LU factorization could be used if the problem is not too large. An alternative approach is to eliminate u^{k+1} from the second equation using the first one (Schur complement reduction), leading to a smaller (order m) linear system of the form

$$[B(I_n + \alpha^{-1}S)^{-1}B^T + \alpha^2 I_m]p^{k+1} = B(I_n + \alpha^{-1}S)^{-1}f^k + \alpha g^k. \tag{2.4}$$

Once the solution p^{k+1} to (2.4) has been computed, the vector u^{k+1} is given by $u^{k+1} = (\alpha I_n + S)^{-1}(f^k - B^T p^{k+1})$. When $S = O$, system (2.4) simplifies to

$$(BB^T + \alpha^2 I_m)p^{k+1} = Bf^k + \alpha g^k, \tag{2.5}$$

and $u^{k+1} = \frac{1}{\alpha}(f^k - B^T p^{k+1})$. If BB^T is sufficiently sparse, system (2.5) could be formed explicitly and solved by a sparse Cholesky factorization. Otherwise, a PCG iteration with a simple preconditioner not requiring access to all the entries of the coefficient matrix $BB^T + \alpha^2 I_m$ could be used. However, when $S \neq O$ the coefficient matrix in (2.4) is generally dense. A nonsymmetric Krylov method could be used to solve (2.4), requiring matrix-vector products with the matrix $B(I_n + \alpha^{-1}S)^{-1}B^T + \alpha^2 I_m$. In turn, this requires solving a linear system of the form $(\alpha I_n + S)v = z$ at each step.

Also note that up to a scaling factor, the coefficient matrix of the coupled system in (2.3) is a normal matrix of the form "identity-plus-skew-symmetric." There are various Lanczos-type methods that can be applied to systems of this kind; see [14, 61] and, more generally, [38]. Other iterative methods for the solution of shifted skew-symmetric systems can be found, e.g., in [47] and [29].

Yet another possibility is to regard (2.3) as a general nonsymmetric system and to use preconditioned GMRES (say). Many of these schemes can benefit form the fact that for even moderate values of $\alpha > 0$, the condition number of $\mathcal{S} + \alpha\mathcal{I}$ is often rather small.

It is important to stress that the linear systems in (2.1) need not be solved exactly. The use of inexact solves was considered in [2] for the positive real case. The upshot is that inexact solves can be used to greatly reduce the cost of each iteration, at the expense of somewhat slower convergence. Typically, in practical implementations, inexact solves result in a much more competitive algorithm. Here we observe that when the alternating scheme is used as a preconditioner for a Krylov method, inexact solves are a natural choice, and there is no theoretical restriction on the accuracy of the inner solves. Inexact solutions are often obtained by iterative methods, leading to an inner-outer scheme; in this case, a flexible solver like FGMRES [52] should be used for the outer iteration. However, inexact solves may also be done by means of incomplete factorizations. In this case, standard GMRES can be used for the outer iteration.

Finally, we note that the scalar matrix $\alpha\mathcal{I}$ in (2.1) could be replaced by a matrix of the form $\alpha\mathcal{F}$, where $\mathcal{F}$ is SPD. This idea, in the context of ADI methods, goes back to Wachspress and Habetler [59]; see also [58, p. 242]. It is straightforward to see that this is equivalent to applying the alternating iteration (2.1) to the symmetrically preconditioned system

$$\hat{\mathcal{A}}\hat{\mathbf{x}} = \hat{\mathbf{b}}, \quad \hat{\mathcal{A}} := \mathcal{F}^{-1/2}\mathcal{A}\mathcal{F}^{-1/2}, \quad \hat{\mathbf{x}} = \mathcal{F}^{1/2}\mathbf{x}, \quad \hat{\mathbf{b}} = \mathcal{F}^{-1/2}\mathbf{b}. \tag{2.6}$$

In this paper we limit ourselves to the case where $\mathcal{F}$ is the $(n+m)\times(n+m)$ diagonal matrix having the ith diagonal entry equal to the ith diagonal entry of $\mathcal{A}$ if this is nonzero, and one otherwise. As we show in the section on numerical experiments, in many cases this simple diagonal preconditioning may considerably improve the rate of convergence.

In the next section we turn to the study of the convergence of the general scheme (2.1), assuming that the solves in (2.2) and (2.3) are performed exactly (rather than approximately, as in an inexact inner-outer setting).

3. Analysis of the stationary iteration. To analyze the convergence of (2.1), we eliminate the intermediate vector $\mathbf{x}^{k+\frac{1}{2}}$ and write the iteration in fixed point form as

$$\mathbf{x}^{k+1} = \mathcal{T}_\alpha \mathbf{x}^k + \mathbf{c}, \tag{3.1}$$

where

$$\mathcal{T}_\alpha := (\mathcal{S} + \alpha\mathcal{I})^{-1}(\alpha\mathcal{I} - \mathcal{H})(\mathcal{H} + \alpha\mathcal{I})^{-1}(\alpha\mathcal{I} - \mathcal{S})$$

is the iteration matrix of the method, and $\mathbf{c} := (\mathcal{S}+\alpha\mathcal{I})^{-1}[\mathcal{I}+(\alpha\mathcal{I}-\mathcal{H})(\mathcal{H}+\alpha\mathcal{I})^{-1}]\mathbf{b}$. The fixed point iteration (3.1) converges for arbitrary initial guesses $\mathbf{x}^0$ and right-hand sides $\mathbf{b}$ to the solution $\mathbf{x} = \mathcal{A}^{-1}\mathbf{b}$ if and only if $\rho(\mathcal{T}_\alpha) < 1$, where $\rho(\mathcal{T}_\alpha)$ denotes the spectral radius of $\mathcal{T}_\alpha$.

It was shown in [2] that when $\mathcal{A}$ is positive real (i.e., $\mathcal{H}$ is SPD), the stationary iteration is unconditionally convergent: $\rho(\mathcal{T}_\alpha) < 1$ for all $\alpha > 0$. However, in the context of generalized saddle point problems (1.2), the matrix $\mathcal{H}$ is only positive semidefinite and generally singular. In this case, the analysis in [2] does not apply. Indeed, for matrices whose symmetric part is positive semidefinite and singular, the alternating iteration is not convergent in general.

However, as the following theorem shows, the alternating iteration converges for a broad class of generalized saddle point problems.

THEOREM 3.1. *Consider problem* (1.2) *and assume that A is positive real, C is symmetric positive semidefinite, and B has full rank. Then the iteration* (3.1) *is unconditionally convergent; that is, $\rho(\mathcal{T}_\alpha) < 1$ for all $\alpha > 0$.*

Proof. The iteration matrix $\mathcal{T}_\alpha$ is similar to

$$\hat{\mathcal{T}}_\alpha = (\alpha\mathcal{I} - \mathcal{H})(\alpha\mathcal{I} + \mathcal{H})^{-1}(\alpha\mathcal{I} - \mathcal{S})(\alpha\mathcal{I} + \mathcal{S})^{-1} = \mathcal{R}\,\mathcal{U},$$

where $\mathcal{R} := (\alpha\mathcal{I}-\mathcal{H})(\alpha\mathcal{I}+\mathcal{H})^{-1}$ is symmetric and $\mathcal{U} := (\alpha\mathcal{I}-\mathcal{S})(\alpha\mathcal{I}+\mathcal{S})^{-1}$ orthogonal (see [37, p. 440]). Now, $\mathcal{R}$ is orthogonally similar to the $(n+m)\times(n+m)$ diagonal matrix

$$\mathcal{D} = \begin{bmatrix} \frac{\alpha-\mu_1}{\alpha+\mu_1} & & & & & & & & \\ & \frac{\alpha-\mu_2}{\alpha+\mu_2} & & & & & & & \\ & & \ddots & & & & & & \\ & & & \frac{\alpha-\mu_n}{\alpha+\mu_n} & & & & & \\ & & & & \frac{\alpha-\nu_1}{\alpha+\nu_1} & & & & \\ & & & & & \frac{\alpha-\nu_2}{\alpha+\nu_2} & & & \\ & & & & & & \ddots & & \\ & & & & & & & \frac{\alpha-\nu_m}{\alpha+\nu_m} & \end{bmatrix},$$

where $\mu_1, \mu_2, \ldots, \mu_n$ are the (positive) eigenvalues of H and $\nu_1, \nu_2, \ldots, \nu_m$ are the (nonnegative) eigenvalues of C. That is, there is an orthogonal matrix $\mathcal{V}$ of order $n+m$ such that

$$\mathcal{V}^T\mathcal{R}\mathcal{V} = \mathcal{D} = \begin{bmatrix} D_1 & O \\ O & D_2 \end{bmatrix},$$

where D_1 and D_2 are diagonal matrices of order n and m, respectively.

Note that

$$\left|\frac{\alpha-\mu_i}{\alpha+\mu_i}\right| < 1 \text{ for } 1 \le i \le n \quad \text{and} \quad \left|\frac{\alpha-\nu_i}{\alpha+\nu_i}\right| \le 1 \text{ for } 1 \le i \le m.$$

It follows that $\mathcal{RU}$ is orthogonally similar to

$$\mathcal{V}^T\mathcal{RUV} = (\mathcal{V}^T\mathcal{RV})(\mathcal{V}^T\mathcal{UV}) = \mathcal{DQ},$$

where $\mathcal{Q} := \mathcal{V}^T\mathcal{UV}$, being a product of orthogonal matrices, is orthogonal. Hence, the iteration matrix $\mathcal{T}_\alpha$ is similar to $\mathcal{DQ}$, and therefore

$$\rho(\mathcal{T}_\alpha) = \rho(\mathcal{DQ}) = \rho(\mathcal{QD}).$$

We claim that $\rho(\mathcal{QD}) < 1$ for all $\alpha > 0$. To show this, partition $\mathcal{Q}$ conformally to $\mathcal{D}$:

$$\mathcal{Q} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}.$$

Then

$$\mathcal{QD} = \begin{bmatrix} Q_{11}D_1 & Q_{12}D_2 \\ Q_{21}D_1 & Q_{22}D_2 \end{bmatrix}.$$

Now, let $\lambda \in \mathbb{C}$ be an eigenvalue of $\mathcal{QD}$ and let $\mathbf{x} \in \mathbb{C}^{n+m}$ be a corresponding eigenvector with $||\mathbf{x}||_2 = 1$. We assume $\lambda \neq 0$, or else there is nothing to prove. We want to show that $|\lambda| < 1$. Clearly, $\mathcal{QD}\mathbf{x} = \lambda\mathbf{x}$ implies $\mathcal{D}\mathbf{x} = \lambda\mathcal{Q}^T\mathbf{x}$ and taking norms:

$$||\mathcal{D}\mathbf{x}||_2 = |\lambda|\,||\mathcal{Q}^T\mathbf{x}||_2 = |\lambda|.$$

Therefore

$$|\lambda|^2 = ||\mathcal{D}\mathbf{x}||_2^2 = \sum_{i=1}^{n}\left(\frac{\alpha-\mu_i}{\alpha+\mu_i}\right)^2 x_i\bar{x}_i + \sum_{i=n+1}^{n+m}\left(\frac{\alpha-\nu_i}{\alpha+\nu_i}\right)^2 x_i\bar{x}_i \le ||\mathbf{x}||_2^2 = 1. \tag{3.2}$$

Hence, the spectral radius of $\mathcal{T}_\alpha$ cannot exceed unity.

To prove that $|\lambda| < 1$ (strictly), we show that there exists at least one i $(1 \le i \le n)$ such that $x_i \neq 0$. Using the assumption that B has full rank, we will show that $x_i = 0$ for all $1 \le i \le n$ implies $\mathbf{x} = \mathbf{0}$, a contradiction. Indeed, if the eigenvector $\mathbf{x}$ is of the form $\mathbf{x} = \left[\begin{smallmatrix} 0 \\ \hat{x} \end{smallmatrix}\right]$ (where $\hat{x} \in \mathbb{C}^m$), the identity $\mathcal{QD}\mathbf{x} = \lambda\mathbf{x}$ becomes

$$\mathcal{QD}\mathbf{x} = \begin{bmatrix} Q_{11}D_1 & Q_{12}D_2 \\ Q_{21}D_1 & Q_{22}D_2 \end{bmatrix}\begin{bmatrix} 0 \\ \hat{x} \end{bmatrix} = \begin{bmatrix} Q_{12}D_2\hat{x} \\ Q_{22}D_2\hat{x} \end{bmatrix} = \begin{bmatrix} 0 \\ \lambda\hat{x} \end{bmatrix} \tag{3.3}$$

so that, in particular, it must be $Q_{12}D_2\hat{x} = 0$. We will prove shortly that Q_{12} has full column rank; hence, it must be $D_2\hat{x} = 0$. But by (3.3) we have $\lambda\hat{x} = Q_{22}D_2\hat{x} = 0$, and since $\lambda \neq 0$ by assumption, it must be $\hat{x} = 0$ (a contradiction, since $\mathbf{x} \neq \mathbf{0}$).

To conclude the proof we need to show that $Q_{12} \in \mathbb{R}^{n\times m}$ has full column rank. Recall that $\mathcal{Q} = \mathcal{V}^T\mathcal{UV}$ with

$$\mathcal{V} = \begin{bmatrix} V_{11} & O \\ O & V_{22} \end{bmatrix},$$

where $V_{11} \in \mathbb{R}^n$ is the orthogonal matrix that diagonalizes $(\alpha I_n - H)(\alpha I_n + H)^{-1}$ and $V_{22} \in \mathbb{R}^m$ is the orthogonal matrix that diagonalizes $(\alpha I_m - C)(\alpha I_m + C)^{-1}$. Recall that the orthogonal matrix $\mathcal{U}$ is given by

$$(\alpha\mathcal{I} - \mathcal{S})(\alpha\mathcal{I} + \mathcal{S})^{-1} = \begin{bmatrix} \alpha I_n - S & -B^T \\ B & \alpha I_m \end{bmatrix} \begin{bmatrix} \alpha I_n + S & B^T \\ -B & \alpha I_m \end{bmatrix}^{-1} = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}.$$

An explicit calculation reveals that

$$U_{12} = -[(\alpha I_n - S)(\alpha I_n + S)^{-1} + I_n]B^T[\alpha I_m + B(\alpha I_n + S)^{-1}B^T]^{-1}.$$

Clearly, -1 cannot be an eigenvalue of the orthogonal matrix $(\alpha I_n - S)(\alpha I_n + S)^{-1}$, hence $(\alpha I_n - S)(\alpha I_n + S)^{-1} + I_n$ is nonsingular. The matrix $\alpha I_m + B(\alpha I_n + S)^{-1}B^T$ is also nonsingular, since $B(\alpha I_n + S)^{-1}B^T$ is positive real. Indeed $(\alpha I_n + S)^{-1}$, being the inverse of a positive real matrix, is itself positive real and since B has full column rank, so is $B(\alpha I_n + S)^{-1}B^T$.

Furthermore,

$$\mathcal{Q} = \mathcal{V}^T\mathcal{U}\mathcal{V} = \begin{bmatrix} V_{11}^T U_{11} V_{11} & V_{11}^T U_{12} V_{22} \\ V_{22}^T U_{21} V_{11} & V_{22}^T U_{22} V_{22} \end{bmatrix}$$

and therefore

$$Q_{12} = V_{11}^T U_{12} V_{22} = -V_{11}^T[(\alpha I_n - S)(\alpha I_n + S)^{-1} + I_n]B^T[\alpha I_m + B(\alpha I_n + S)^{-1}B^T]^{-1}V_{22},$$

showing that Q_{12} has full column rank since V_{11}^T and V_{22} are orthogonal and B^T has full column rank. This completes the proof. □

REMARK 3.1. *It is easy to see that there is a unique splitting $\mathcal{A} = \mathcal{M} - \mathcal{N}$ with $\mathcal{M}$ nonsingular such that the iteration matrix $\mathcal{T}_\alpha$ is the matrix induced by that splitting, i.e., $\mathcal{T}_\alpha = \mathcal{M}^{-1}\mathcal{N} = \mathcal{I} - \mathcal{M}^{-1}\mathcal{A}$. An easy calculation shows that*

$$\mathcal{M} \equiv \mathcal{M}_\alpha = \tfrac{1}{2\alpha}(\mathcal{H} + \alpha\mathcal{I})(\mathcal{S} + \alpha\mathcal{I}). \tag{3.4}$$

It is therefore possible to rewrite the iteration (2.1) *in* correction form:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathcal{M}_\alpha^{-1}\mathbf{r}^k, \quad \mathbf{r}^k = \mathbf{b} - \mathcal{A}\mathbf{x}^k.$$

This will be useful when we consider Krylov subspace acceleration.

The restriction in Theorem 3.1 that A be positive real is not essential. If A is only semipositive real (singular), the alternating iteration (2.1) is still well defined, but it may happen that $\rho(\mathcal{T}_\alpha) = 1$ for *all* values of $\alpha > 0$. A simple example with $n = 2$, $m = 1$ is given by

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

Nevertheless, a simple modification of the basic algorithm yields a convergent iteration. To this end, recall that $\rho(\mathcal{T}_\alpha) \le 1$ for all $\alpha > 0$; see (3.2). Also, note that $1 \notin \sigma(\mathcal{T}_\alpha)$ since $\mathcal{A}$ is nonsingular. Let $\beta \in (0, 1)$ be a parameter; then the matrix $(1 - \beta)\mathcal{I} + \beta\mathcal{T}_\alpha$ has spectral radius less than 1 for all $\alpha > 0$. Indeed, the eigenvalues of $(1 - \beta)\mathcal{I} + \beta\mathcal{T}_\alpha$ are of the form $1 - \beta + \beta\lambda$, where $\lambda \in \sigma(\mathcal{T}_\alpha)$. It is easy to see that since $|\lambda| \le 1$ and $\lambda \ne 1$, all the quantities $1 - \beta + \beta\lambda$ have magnitude strictly less

than 1. This trick is routinely used in the solution of singular systems and Markov chains; see, e.g., [40].

Thus, for any choice of the initial guess $\hat{\mathbf{x}}^0 = \mathbf{x}^0$, the sequence $\{\hat{\mathbf{x}}^k\}$ defined by

$$\hat{\mathbf{x}}^{k+1} = (1-\beta)\hat{\mathbf{x}}^k + \beta\mathbf{x}^{k+1} = (1-\beta)\hat{\mathbf{x}}^k + \beta(\mathcal{T}_\alpha\hat{\mathbf{x}}^k + \mathbf{c})$$

($k = 0, 1, \ldots$) converges to the unique solution of problem (1.2) for all $\beta \in (0, 1)$ and all $\alpha > 0$. In this way, the alternating iteration is applicable to any generalized saddle point problem. The presence of the parameter β, unfortunately, adds another complication to the method. Numerical experiments suggest that a value of β slightly less than 1, like $\beta = 0.99$, should be used. When Krylov subspace acceleration is used, however, there is no need to use this technique (that is, one can use $\beta = 1$ even when H is singular).

Under the assumptions of Theorem 3.1, the asymptotic rate of convergence of the alternating iteration is governed by the spectral radius of $\mathcal{T}_\alpha$, so it makes sense to try to choose α so as to make $\rho(\mathcal{T}_\alpha)$ as small as possible. In general, finding such a value $\alpha = \alpha_{\text{opt}}$ is a difficult problem. Some results in this direction can be found in [2, 3, 6]. The results in [2] yield an expression of the optimal α for the case of $\mathcal{A}$ positive real, too restrictive in our setting where $\mathcal{H}$ is usually singular.

Of course, choosing α so as to minimize the spectral radius of the iteration matrix is not necessarily the best choice when the algorithm is used as a preconditioner for a Krylov subspace method. Remarkably, it can be shown that for certain problems the alternating iteration results in an h-independent preconditioner for GMRES when α is chosen sufficiently small, corresponding to a spectral radius very close to 1; see [6] and the numerical experiments in section 5.1 below.

Also, minimizing the spectral radius or even the number of GMRES iterations does not imply optimal performance in terms of CPU time. Indeed, the efficient implementation of the method almost invariably requires that the two linear systems (2.2) and (2.3) be solved inexactly. Clearly, the choice of α will influence the cost of performing the two solves. Indeed, "large" values of α will make the iterative solution of (2.2) and (2.3) easy; on the other hand, it is clear from (3.2) that the nonzero eigenvalues of $\mathcal{T}_\alpha$ approach 1 as $\alpha \to \infty$ (and also as $\alpha \to 0$), and convergence of the outer iteration slows down. Hence, there is a trade-off involved. If we define the "optimal" value of α as the one that minimizes the total amount of work needed to compute an approximate solution, this will not necessarily be the same as the α that minimizes the number of (outer) iterations. Overall, the analytic determination of such an optimal value for α appears to be daunting.

4. Krylov subspace acceleration. Even with the optimal choice of α, the convergence of the stationary iteration (2.1) is typically too slow for the method to be competitive. For this reason we propose using a nonsymmetric Krylov subspace method like GMRES, or its restarted version GMRES(m), to accelerate the convergence of the iteration.

It follows from Remark 3.1 that the linear system $\mathcal{A}\mathbf{x} = \mathbf{b}$ is equivalent to (i.e., has the same solution as) the linear system

$$(\mathcal{I} - \mathcal{T}_\alpha)\mathbf{x} = \mathcal{M}_\alpha^{-1}\mathcal{A}\mathbf{x} = \mathbf{c},$$

where $\mathbf{c} = \mathcal{M}_\alpha^{-1}\mathbf{b}$. This equivalent (left-preconditioned) system can be solved with GMRES. Hence, the matrix $\mathcal{M}_\alpha$ can be seen as a *preconditioner* for GMRES. Equivalently, we can say that GMRES is used to accelerate the convergence of the alternating iteration applied to $\mathcal{A}\mathbf{x} = \mathbf{b}$.

Note that as a preconditioner we can use

$$\mathcal{M}_\alpha = (\mathcal{H} + \alpha\mathcal{I})(\mathcal{S} + \alpha\mathcal{I})$$

instead of the expression given in (3.4), since the factor $\frac{1}{2\alpha}$ has no effect on the preconditioned system. Application of the alternating preconditioner within GMRES requires solving a linear system of the form $\mathcal{M}_\alpha \mathbf{z} = \mathbf{r}$ at each iteration. This is done by first solving

$$(\mathcal{H} + \alpha\mathcal{I})\mathbf{v} = \mathbf{r} \tag{4.1}$$

for $\mathbf{v}$, followed by

$$(\mathcal{S} + \alpha\mathcal{I})\mathbf{z} = \mathbf{v}. \tag{4.2}$$

The GMRES method can also be applied to the right-preconditioned system $\mathcal{A}\mathcal{M}_\alpha^{-1}\mathbf{y} = \mathbf{b}$ where $\mathbf{y} = \mathcal{M}_\alpha \mathbf{x}$. Note that $\mathcal{M}_\alpha^{-1}\mathcal{A}$ and $\mathcal{A}\mathcal{M}_\alpha^{-1}$ are similar and therefore have the same eigenvalues. In principle, the convergence behavior of GMRES can be different depending on whether left- or right-preconditioning is being used, but in our numerical experiments we noticed little or no difference.

Under the assumptions of Theorem 3.1, since $\mathcal{M}_\alpha^{-1}\mathcal{A} = \mathcal{I} - \mathcal{T}_\alpha$ it is readily seen that for all $\alpha > 0$ the eigenvalues of the preconditioned matrix $\mathcal{M}_\alpha^{-1}\mathcal{A}$ (or of $\mathcal{A}\mathcal{M}_\alpha^{-1}$) are entirely contained in the open disk of radius 1 centered at $(1, 0)$. In particular, the preconditioned matrix is positive stable. The smaller the spectral radius of $\mathcal{T}_\alpha$, the more clustered the eigenvalues of the preconditioned matrix (around 1); a clustered spectrum often translates in rapid convergence of GMRES.

If a matrix is positive real, then it is positive stable; the converse, however, is not true. A counterexample is given by a matrix of the form

$$A = \begin{bmatrix} 1 & 0 \\ a & 1 \end{bmatrix},$$

where a is any real number with $|a| \geq 2$. The question then arises whether $\mathcal{M}_\alpha^{-1}\mathcal{A}$ (or $\mathcal{A}\mathcal{M}_\alpha^{-1}$) is positive real, for in this case the convergence of GMRES(m) would be guaranteed for all restarts m; see [19] and [53, p. 866]. Unfortunately, this is not true in general. However, when A is SPD and $C = O$ we can prove that the preconditioned matrix is positive real provided that α is sufficiently large.

THEOREM 4.1. *Assume A is SPD, $C = O$, and B has full rank. Then there exists $\alpha^* > 0$ such that $\mathcal{M}_\alpha^{-1}\mathcal{A}$ is positive real for all $\alpha > \alpha^*$. An analogous result holds for the right-preconditioned matrix, $\mathcal{A}\mathcal{M}_\alpha^{-1}$.*

Proof. For brevity, we prove the theorem only for the left-preconditioned matrix; the proof for the right-preconditioned one is similar. Up to a positive scalar multiple, the symmetric part of the preconditioned matrix $\mathcal{M}_\alpha^{-1}\mathcal{A}$ is given by

$$\mathcal{B} = (\mathcal{S} + \alpha\mathcal{I})^{-1}(\mathcal{H} + \alpha\mathcal{I})^{-1}\mathcal{A} + \mathcal{A}^T(\mathcal{H} + \alpha\mathcal{I})^{-1}(\alpha\mathcal{I} - \mathcal{S})^{-1}$$

(where we have used the fact that $\mathcal{S}^T = -\mathcal{S}$). This matrix is congruent to

$$(\mathcal{S} + \alpha\mathcal{I})\mathcal{B}(\mathcal{S} + \alpha\mathcal{I})^T = (\mathcal{H} + \alpha\mathcal{I})^{-1}\mathcal{A}(\alpha\mathcal{I} - \mathcal{S}) + (\mathcal{S} + \alpha\mathcal{I})\mathcal{A}^T(\mathcal{H} + \alpha\mathcal{I})^{-1},$$

which, in turn, is congruent to the inverse-free matrix

$$\mathcal{Z} = \mathcal{A}(\alpha\mathcal{I} - \mathcal{S})(\mathcal{H} + \alpha\mathcal{I}) + (\mathcal{H} + \alpha\mathcal{I})(\mathcal{S} + \alpha\mathcal{I})\mathcal{A}^T.$$

A direct calculation shows that

$$\mathcal{Z} = \begin{bmatrix} Z_\alpha & -2\alpha AB^T \\ -2\alpha BA & 2\alpha BB^T \end{bmatrix},$$

where

$$Z_\alpha := 2\alpha A^2 + 2\alpha B^T B + 2\alpha^2 A + B^T BA + AB^T B.$$

We want to show that $\mathcal{Z}$ is SPD for sufficiently large α. To this end, we observe that $\mathcal{Z}$ can be split as

$$\mathcal{Z} = 2\begin{bmatrix} \alpha A^2 & -\alpha AB^T \\ -\alpha BA & \alpha BB^T \end{bmatrix} + \begin{bmatrix} M_\alpha & O \\ O & O \end{bmatrix}, \tag{4.3}$$

where

$$M_\alpha := 2\alpha^2 A + 2\alpha B^T B + B^T BA + AB^T B.$$

The first matrix on the right-hand side of (4.3) is symmetric positive semidefinite, since

$$\begin{bmatrix} \alpha A^2 & -\alpha AB^T \\ -\alpha BA & \alpha BB^T \end{bmatrix} = \begin{bmatrix} \alpha A & O \\ -\alpha B & I_m \end{bmatrix} \begin{bmatrix} \alpha^{-1} I_n & O \\ O & O \end{bmatrix} \begin{bmatrix} \alpha A & -\alpha B^T \\ O & I_m \end{bmatrix}.$$

Next, we observe that

$$M_\alpha = \alpha(2B^T B + 2\alpha A) + (B^T BA + AB^T B)$$

is similar to a matrix of the form $\alpha I_n + W$, where $W = W^T$ is generally indefinite. This matrix can be made SPD by taking α sufficiently large. Specifically, M_α is SPD for all $\alpha > \alpha^*$, where

$$\alpha^* = -\lambda_{\min}(B^T BA + AB^T B)$$

(note that $B^T BA + AB^T B$ is generally indefinite). Hence, for $\alpha > \alpha^*$ the matrix $\mathcal{Z}$ is the sum of two symmetric positive semidefinite matrices; therefore, it is itself symmetric positive semidefinite. Finally, it must be nonsingular for all $\alpha > \alpha^*$ (and therefore positive definite). Indeed, it is clear from (4.3) that when M_α is positive definite, any null vector of $\mathcal{Z}$ must be of the form

$$\mathbf{x} = \begin{bmatrix} 0 \\ \hat{x} \end{bmatrix}, \quad \text{where } \hat{x} \in \mathbb{R}^m.$$

But then

$$\mathcal{Z}\mathbf{x} = 2\begin{bmatrix} \alpha A^2 & -\alpha AB^T \\ -\alpha BA & \alpha BB^T \end{bmatrix} \begin{bmatrix} 0 \\ \hat{x} \end{bmatrix} = \begin{bmatrix} -2\alpha AB^T \hat{x} \\ 2\alpha BB^T \hat{x} \end{bmatrix},$$

which cannot be zero unless $\hat{x} = 0$, since B^T has full column rank and A is nonsingular. Hence $\mathcal{Z}$ has no nontrivial null vectors for $\alpha > \alpha^*$. This shows that the symmetric part of the preconditioned matrix is SPD for all $\alpha > \alpha^*$, since it is congruent to a matrix which is SPD for all such values of α. □

It is worth mentioning that in all cases that we were able to check numerically, we found the symmetric part of the preconditioned operator to be positive definite already for rather small values of α.

More refined bounds and clustering results for the eigenvalues of $\mathcal{M}_\alpha^{-1}\mathcal{A}$ can be found in [55].

5. Numerical experiments. In this section we present a sample of numerical experiments conducted in order to assess the effectiveness of the alternating algorithm (2.1) both as a stationary iterative scheme and as a preconditioner for GMRES. All experiments were performed in Matlab. Our codes have not been optimized for highest efficiency and therefore we do not report timings, but we do provide cost estimates for some of the test problems. We think that the results of the experiments presented here provide evidence of the fact that our approach is worth further consideration.

We target matrices from different application areas, but mostly from PDE problems. In all our runs we used a zero initial guess and stopped the iteration when the relative residual had been reduced by at least six orders of magnitude (i.e., when $||\mathbf{b} - \mathcal{A}\mathbf{x}^k||_2 \leq 10^{-6}||\mathbf{b}||_2$).

5.1. Second order equations in first order system form. Let $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ be a bounded open set. Here we consider the numerical solution of boundary value problems for the following second order elliptic PDE:

$$-\nabla \cdot (K \nabla p) = g \quad \text{in } \Omega, \tag{5.1}$$

where $K = K(\mathbf{r})$ is a strictly positive function or tensor for $\mathbf{r} \in \bar{\Omega}$ and $g(\mathbf{r})$ is a given forcing term. Equation (5.1) is complemented by appropriate boundary conditions.

The PDE (5.1) is equivalent to the following system of two first order PDEs:

$$\begin{cases} K^{-1}\,\mathbf{u} - \nabla p = \mathbf{0}, \\ -\nabla \cdot \mathbf{u} = g. \end{cases} \tag{5.2}$$

Discretization of these equations leads to large sparse linear systems in saddle point form (1.2).

We begin with the simplest possible case, namely, Poisson's equation on the unit square:

$$-\Delta p = -\nabla \cdot (\nabla p) = g \quad \text{in } \Omega = [0, 1] \times [0, 1].$$

This corresponds to taking $K \equiv 1$ in (5.1). We discretize form (5.2) of the problem using finite differences with a forward difference for the gradient and a backward difference for the divergence. Using an $N \times N$ uniform grid with mesh size $h = \frac{1}{N+1}$ results in a linear system of type (1.2) with $n = 2N^2$ and $m = N^2$, for a total system size of $3N^2$ equations in as many unknowns.

As shown in [6], for this model problem Fourier analysis at the continuous (differential operator) level can be used to completely analyze the spectrum of the iteration operator $\mathcal{T}_\alpha$. This allows us to find the optimal value α_{opt} of the parameter as a function of h, showing that the spectral radius for the stationary iteration (2.1) behaves as $1 - c\sqrt{h}$ as $h \to 0$. The optimal value α_{opt} itself behaves as $h^{-\frac{1}{2}}$ as $h \to 0$. More interestingly, the spectral analysis in [6] indicates that when GMRES acceleration is used, a better choice is to use a small value of α, for it can be shown that for $\alpha \in (0, 1)$ the eigenvalues of the preconditioned matrix lie in two intervals which depend on α, but do not depend on h, resulting in h-independent convergence. In particular, α can always be chosen so as to have convergence within 2–3 iterations, uniformly in h.

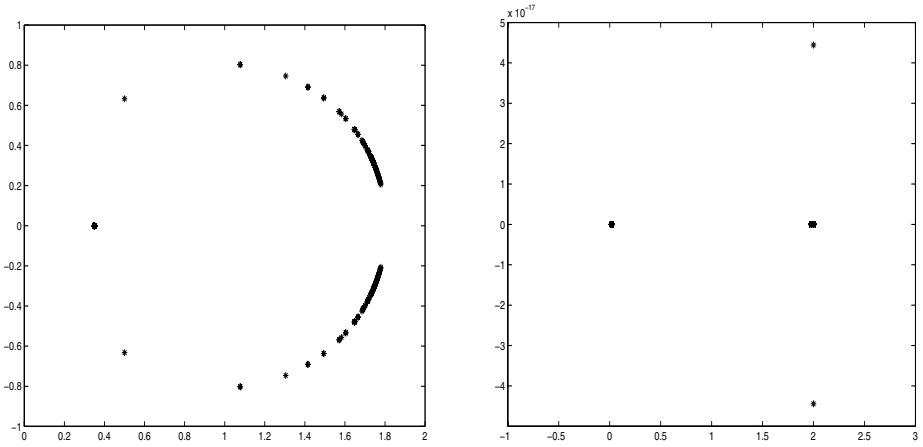
This behavior is illustrated in Table 5.1. We take the forcing term to be the function $g(x, y) = \sin \pi x \sin \pi y$ and we impose Neumann boundary conditions for $x = 0$, $x = 1$, and homogeneous Dirichlet boundary conditions for $y = 0$, $y = 1$. The numerical results are in agreement with the theoretical analysis. In particular, note

TABLE 5.1
Two-dimensional Poisson's equation. Comparison of iterative scheme optimized as an iterative solver, full GMRES without preconditioner, GMRES with the optimized iterative scheme as a preconditioner and iterative scheme optimized for GMRES.

		GMRES		
h	Iterative	No Prec.	Preconditioned	Optimized
1/10	66	54	14	2
1/25	103	140	19	2
1/50	146	286	25	2
1/100	207	574	34	2

that convergence is attained in two steps (independent of h) when the iteration is optimized for GMRES acceleration. Here we used $\alpha = 0.001$, but the behavior of the preconditioned iteration is not very sensitive to the choice of $\alpha \in (0, 1)$.

In Figure 5.1 we display the eigenvalues of the preconditioned matrix $\mathcal{M}_\alpha^{-1}\mathcal{A}$ in the case of $h = \frac{1}{10}$ for two values of α. On the left we used the value $\alpha = \alpha_{\rm opt}$ that minimizes the spectral radius, which is given by $\rho(\mathcal{T}_{\alpha_{\rm opt}}) = 0.8062$. On the right we used $\alpha = 0.01$, showing the clustering near 0 and 2 predicted by the theory developed in [6]. Now the spectral radius of the iteration matrix is very close to 1. The cluster near 0 contains $m = 81$ eigenvalues, the one near 2 the remaining $n = 162$. It should be noted that the (tiny) imaginary part in Figure 5.1(b) is due to round-off error, since the eigenvalues are real for small α; see [6, 55].



(a) Spectrum for $\alpha = \alpha_{\rm opt}$ (b) Spectrum for $\alpha = 0.01$

FIG. 5.1. *Eigenvalues of preconditioned matrices for the Poisson problem on a* 10×10 *grid.*

Next we consider a somewhat harder problem, namely, the anisotropic equation

$$-100\, p_{xx} - p_{yy} = g \quad \text{in } \Omega = [0, 1] \times [0, 1].$$

Since this problem has constant coefficients, the technique used in [6] for Poisson's equation can be used to optimize the method. The results in Table 5.2 show that the anisotropy in the coefficients drastically decreases the rate of convergence. However, in this case there is an easy fix: as the results reported in Table 5.3 show, it is enough to apply the scaling (2.6) to restore the effectiveness of the solver. We note that a similar scaling has been used in [13] in a somewhat different context.

TABLE 5.2
Results for two-dimensional problem with anisotropic coefficients.

		GMRES		
h	Iterative	No Prec.	Preconditioned	Optimized
1/10	709	186	34	29
1/25	> 1000	651	44	31
1/50	> 1000	> 1000	52	31
1/100	> 1000	> 1000	59	31

TABLE 5.3
Results for two-dimensional problem with anisotropic coefficients, diagonally scaled.

		GMRES		
h	Iterative	No Prec.	Preconditioned	Optimized
1/10	138	100	15	2
1/25	210	344	17	2
1/50	292	> 500	22	2
1/100	400	> 500	29	2

Finally, we consider a more difficult problem with large jumps in the coefficients K. The system is discretized using a discontinuous Galerkin finite element scheme. This radiation diffusion problem arises in a nuclear engineering application and was supplied to us by James Warsa of Los Alamos National Laboratory. For more details, see [60] and the references therein. For this problem $n = 2592$, $m = 864$, $n+m = 3456$, and $\mathcal{A}$ contains 93,612 nonzero entries. Here $C \neq O$ (and indeed it is SPD).

The results for this problem are presented in Table 5.4, where the entries in the first row correspond to GMRES with diagonal preconditioning (2.6). We give results for full GMRES and for restarted GMRES with restart every 20 steps. Here we cannot apply Fourier analysis to optimize the choice of α as we did in the constant coefficient cases. Therefore, we experiment with different values of α. While the fastest convergence rate for the stationary iterative methods correspond to $\alpha = 0.25$, a somewhat bigger α works best if the method is used as a preconditioner for GMRES. In any case the method is not overly sensitive to the choice of α when GMRES acceleration is used. We stress here again the importance of the diagonal scaling (2.6), which results in a reduction by a factor of two in the number of iterations for this problem.

5.2. Stokes and Oseen problems. In this section we present a few results for discretizations of Stokes and Oseen problems. Recall that the Stokes system is

$$\begin{cases} -\Delta \mathbf{u} + \nabla p = \mathbf{f}, \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \tag{5.3}$$

in $\Omega \subset \mathbb{R}^d$, together with suitable boundary conditions. Here $\mathbf{u}$ denotes the velocity vector field and p the pressure scalar field. Discretization of (5.3) using stabilized finite elements leads to saddle point problems of the type (1.2) with a symmetric positive definite A and a symmetric positive semidefinite C.

The Oseen equations are obtained when the steady-state Navier–Stokes equations are linearized by Picard iteration:

$$\begin{cases} -\nu \Delta \mathbf{u} + (\mathbf{v} \cdot \nabla)\, \mathbf{u} + \nabla p = \mathbf{f}, \\ \nabla \cdot \mathbf{u} = 0\,. \end{cases} \tag{5.4}$$

Here the vector field $\mathbf{v}$ is the approximation of $\mathbf{u}$ from the previous Picard iteration. The parameter $\nu > 0$ represents viscosity. Various approximation schemes can be used

TABLE 5.4
Results for discontinuous radiation diffusion equations.

α	Iterative	GMRES	GMRES(20)
–		697	> 1000
0.1	750	155	239
0.2	375	100	113
0.25	257	97	101
0.3	304	89	100
0.4	404	85	95
0.5	504	85	100
0.6	604	81	98
0.7	704	80	108
0.8	805	80	115
0.9	905	80	120
1.0	1005	82	135

to discretize the Oseen problem (5.4) leading to a generalized saddle point system of type (1.2). Now the A block corresponds to a discretization of the convection-diffusion operator $L[\mathbf{u}] := -\nu\Delta\mathbf{u} + (\mathbf{v}\cdot\nabla)\,\mathbf{u}$. It is nonsymmetric, but for conservative discretizations, the symmetric part is positive definite.

We generated several test problems using the IFISS software package written by Howard Elman, Alison Ramage, and David Silvester. We used this package to generate discretizations of leaky lid driven cavity problems for both the Stokes and Oseen equations. The discretization used is stabilized Q1-P0 finite elements. In all cases the default value of the stabilization parameter ($\beta = 0.25$) was used. It should be mentioned that the matrices generated by this package are actually singular, since B has rank $m - 2$. This does not cause any difficulty to the iterative solvers considered here. In particular, even if $\lambda = 1$ is now an eigenvalue of the iteration matrix $\mathcal{T}_\alpha = \mathcal{I} - \mathcal{M}_\alpha^{-1}\mathcal{A}$, the stationary iteration is still convergent, with a rate of convergence governed by $\gamma(\mathcal{T}_\alpha) := \max\{\,|\lambda|;\, \lambda \in \sigma(\mathcal{T}_\alpha),\, \lambda \neq 1\}$.

For the Stokes problem we used a 16×16 grid. For the Oseen problem we used two grids, 16×16 and 32×32. The first grid corresponds to $n = 578$ and $m = 256$, for a total of 834 unknowns. For the second grid $n = 2178$ and $m = 1024$, for a total of 3202 unknowns. Two values of the viscosity parameter were used for the Oseen problems, $\nu = 0.01$ and $\nu = 0.001$. We experiment with both full GMRES and GMRES(20). Diagonal scaling (2.6) greatly improves the rate of convergence in all cases, and it is used throughout.

Table 5.5 contains results for the Stokes problem with both exact and inexact solves. Although there is no value of α that yields convergence in two steps, the alternating iteration is able to significantly improve the convergence of GMRES. Note that the behavior of the preconditioned iteration is not overly sensitive to the choice of α; in contrast, the rate of convergence of the stationary iteration without GMRES acceleration depends strongly on α. Since the (average) cost of a preconditioned GMRES(20) iteration is approximately three times the cost of an unpreconditioned iteration, the preconditioner allows for a saving of about a factor of two over unpreconditioned GMRES(20), when using the "best" values of α. Better results are obtained with inexact solves corresponding to incomplete factorizations. We used drop tolerance-based incomplete Cholesky for the first system in (2.1) and ILU for the second one. In both cases the drop tolerance was set to $tol = 0.05$. For $\alpha \geq 0.1$ the incomplete Cholesky factor of $\mathcal{H} + \alpha\mathcal{I}$ is very sparse, with about 25% of the nonzeros in the coefficient matrix itself. The ILU factors of $\mathcal{S} + \alpha\mathcal{I}$, for this particular example,

TABLE 5.5
Results for Stokes problem.

	Exact solves			Inexact solves	
α	Iterative	GMRES	GMRES(20)	GMRES	GMRES(20)
–		103	194		
0.01	> 1000	101	205	210	> 500
0.1	801	53	60	58	62
0.2	135	33	36	35	36
0.3	78	29	30	32	30
0.4	107	29	34	33	35
0.5	134	30	39	37	41
0.6	137	32	47	40	48
0.7	165	35	51	42	54
0.8	222	37	58	44	59
0.9	250	39	63	45	64
1.0	277	42	67	46	68

TABLE 5.6
Results for Oseen problem on 16×16 *grid,* $\nu = 0.01$.

	Exact solves			Inexact solves	
α	Iterative	GMRES	GMRES(20)	GMRES	GMRES(20)
–		353	> 500		
0.01	> 1000	105	268	179	> 500
0.1	> 1000	58	66	163	> 500
0.2	> 1000	38	39	107	> 500
0.3	> 1000	29	29	82	166
0.4	842	25	25	70	114
0.5	474	23	23	61	89
0.6	301	22	22	50	62
0.7	203	23	23	43	52
0.8	149	23	23	40	50
0.9	157	24	25	39	49
1.0	170	26	27	40	55
1.1	183	27	30	37	46
1.2	197	29	33	38	51

have around 35% of the nonzeros in the complete factors. As a result, the cost of applying the preconditioner is reduced by about a factor of four (from 62.6×10^3 to 15.4×10^3 operations, per iteration). It can be seen that the rate of convergence deteriorates only slightly. This deterioration is more than compensated by the lower cost per iteration. Moreover, the set-up cost goes down from 292×10^3 operations for the complete factorizations to 81×10^3 for the incomplete ones. Compared to the exact case, the overall reduction in the total number of operations is more than a factor of two for α between 0.4 and 1, while total storage for the preconditioner is reduced by almost a factor of three. Also note that with inexact solves, the (average) cost of a preconditioned GMRES(20) iteration is approximately one and a half times the cost of an unpreconditioned iteration. Hence, for the best values of α, the preconditioner results in a reduction of the cost of GMRES(20) by more than a factor of four.

Table 5.6 contains experimental results for the Oseen problem on the small (16×16) grid with viscosity $\nu = 0.01$. The results for GMRES with diagonal scaling (2.6), reported in the first row, indicate that the Oseen problem is harder than the Stokes problem. Here we see a surprising result: while the stationary iteration tends to converge more slowly than for the Stokes problem, the preconditioned GMRES iteration now tends to converge faster. We think this could be due to the fact

TABLE 5.7
Results for Oseen problem on 16×16 *grid,* $\nu = 0.001$.

	Exact solves			Inexact solves	
α	Iterative	GMRES	GMRES(20)	GMRES	GMRES(20)
–		616	> 1000		
0.01	> 1000	69	177	162	> 1000
0.1	> 1000	42	55	149	> 1000
0.2	> 1000	32	37	131	> 1000
0.3	> 1000	22	28	112	> 1000
0.4	> 1000	22	22	101	> 1000
0.5	> 1000	22	22	90	> 1000
0.6	> 1000	21	21	86	> 1000
0.7	965	21	21	70	664
0.8	713	21	21	76	> 1000
0.9	552	21	21	71	275
1.0	444	22	22	60	166
1.1	364	22	23	53	228
1.2	302	23	24	54	108
1.5	239	25	29	53	137
2.0	286	31	42	56	151

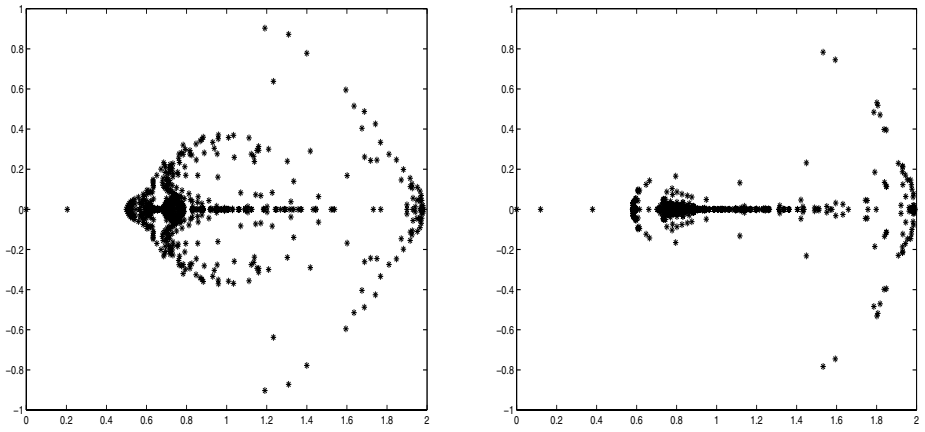
that the coefficient matrix has a more substantial skew-symmetric part in this case, and preconditioning with the (shifted) skew-symmetric part becomes more effective. Now GMRES(20) does not converge within 500 iterations without preconditioning. Full GMRES requires about 5.7 times more flops than the stationary iteration with $\alpha = 0.8$, and about 17 times more than the preconditioned iteration. Note that this estimate includes the set-up time for the preconditioner. The results obtained with inexact solves (by incomplete factorization) show some deterioration (about a factor of two for the "best" α) in convergence rates. This deterioration is more than compensated by the reduced cost of each preconditioned iteration.

In Table 5.7 we report results for the Oseen problem on the 16×16 grid and a viscosity parameter $\nu = 0.001$. Generally speaking, the Oseen problem becomes harder to solve as the viscosity gets smaller; see the results for diagonally scaled GMRES, and for the stationary iteration. However, the combination of the iteration and GMRES acceleration results in even faster convergence than in the previous case of $\nu = 0.01$. In Figure 5.2 we display the eigenvalues of the preconditioned matrix corresponding to the Oseen problem on the 16×16 grid. The plot on the left corresponds to a viscosity $\nu = 0.01$ and the one on the right to $\nu = 0.001$; we used the values of α that resulted in the smallest number of preconditioned GMRES iterations ($\alpha = 0.6$ and $\alpha = 0.8$, respectively). Note the stronger clustering of the spectrum for the case with $\nu = 0.001$.

Unfortunately, this apparent robustness with respect to ν is lost as soon as the exact solves in (2.1) are replaced by inexact solves by incomplete factorization, especially with restarted GMRES. The same value of the drop tolerance $tol = 0.05$ was used in all cases. Whether it is possible to solve the inner problems inexactly and still preserve robustness with respect to ν remains an open question.

Finally, in Table 5.8 we present results for the Oseen problem with $\nu = 0.001$ on the finer grid. The preconditioned GMRES iteration appears to be fairly robust with respect to the mesh size h and the viscosity parameter ν when exact solves are used.

5.3. A problem with singular A. Finally, we consider a saddle point problem arising in geophysics and supplied to us by Eldad Haber of Emory University; see [28, 33, 34]. In this application the submatrix A is symmetric positive semidefinite



(a) Spectrum for $\nu = 0.01$ (b) Spectrum for $\nu = 0.001$

FIG. 5.2. *Eigenvalues of preconditioned matrices for the Oseen problem on a* 16×16 *grid.*

TABLE 5.8
Results for Oseen problem with exact solves on 32×32 *grid,* $\nu = 0.001$.

α	Iterative	GMRES	GMRES(20)
–		> 1000	> 1000
0.1	> 1000	52	58
0.2	> 1000	36	38
0.3	> 1000	32	32
0.4	> 1000	29	30
0.5	> 1000	28	29
0.6	> 1000	28	34
0.7	> 1000	28	40
0.8	> 1000	39	42
0.9	757	31	44
1.0	574	33	47
1.5	684	43	55
2.0	864	53	121

and singular. In the example at hand $n = 1241$, $m = 729$, $n + m = 1970$, and $\mathcal{A}$ contains 25,243 nonzeros. The A block has $\text{rank}(A) = 876$. In this problem, $C = O$.

We present results for this problem in Table 5.9. Diagonal scaling (2.6) drastically improves the convergence of the preconditioned iterations. However, the convergence of the stationary iteration (2.1) without GMRES acceleration remains extremely slow. Likewise for GMRES with no preconditioning or diagonal preconditioning alone. The results with inexact solves in Table 5.9 were obtained by replacing the exact solve with no-fill incomplete factorizations, IC(0) and ILU(0). Again we see a deterioration in convergence rates, but each iteration is now far cheaper than in the case of exact solves, resulting in huge savings. When $\alpha = 0.3$ (but similar results hold for all the other values of α in the table), the Cholesky factorization of $\mathcal{H} + \alpha\mathcal{I}$ requires 1.7×10^6 operations using a minimum degree ordering, resulting in a triangular factor with 31.5×10^3 nonzeros. The complete factorization of $\mathcal{S} + \alpha\mathcal{I}$ (using minimum degree as the initial ordering) costs a staggering 207×10^6 operations with a total number of nonzeros in the factors exceeding 837×10^3. In contrast, the IC(0) factorization of $\mathcal{H} + \alpha\mathcal{I}$ only required 17.6×10^3 operations and resulted in an incomplete Cholesky factor with just 5.2×10^3 nonzeros; the ILU(0) factorization of $\mathcal{S} + \alpha\mathcal{I}$ took

TABLE 5.9
Results for geophysics problem with singular A.

	Exact solves			Inexact solves	
α	Iterative	GMRES	GMRES(30)	GMRES	GMRES(30)
–		> 500	> 500		
0.1	$> 1{,}000$	49	60	85	177
0.2	$> 1{,}000$	42	42	68	110
0.3	$> 1{,}000$	44	62	60	112
0.4	$> 1{,}000$	48	92	58	93
0.5	$> 1{,}000$	51	98	62	104

76.1×10^3 operations and resulted in a total of 21.4×10^3 nonzeros in the incomplete factors.

6. Conclusions and future work. In this paper we have studied the extension of the alternating method of [2] to generalized saddle point problems. Because these linear systems have coefficient matrices with singular symmetric part, they are not positive real. Thus, the convergence analysis carried out in [2] for the positive real case does not apply, and convergence has to be established using different arguments from those used in [2]. Other approaches to studying convergence have been proposed recently in [3] and [6]; see also [8] and [55].

Rather than used as a stand-alone solver, the stationary iteration is best used as a preconditioner for a nonsymmetric Krylov subspace method, such as GMRES. Here we have established theoretical properties of the preconditioned matrices that were relevant for restarted GMRES, at least from a qualitative point of view.

Our numerical experiments with test matrices from several different applications suggest that the combination of GMRES and the alternating iteration is fairly robust, and not overly sensitive to the choice of the parameter α. As demonstrated already in [6] for some model problems, there are important examples of systems of PDEs where the combination of iteration (2.1) with an appropriate choice of the optimization parameter α and GMRES acceleration results in an h-independent solver, or with a weak dependence on h.

Our numerical experiments show that diagonal scaling (2.6) greatly improves the convergence of the outer iteration. We have also performed some experiments with inexact solves. For several of our test problems, the rate of convergence suffered relatively little deterioration, leading to a reduction in overall costs in many cases. However, we also found problems where inexactness in the inner solves resulted in slow convergence, at least when incomplete factorizations were used.

Future work should focus on developing efficient implementations of the algorithm, with particular attention to the problem of striking a balance between the rate of convergence of the outer (preconditioned) iteration, and the amount of work spent performing the inner (inexact) solves. Here we have presented a few results using incomplete factorizations, but iterative methods may be a better (more flexible) option. For the Oseen equations with small viscosity parameter ν, it may be difficult to find inexact inner solves that do not lead to a serious deterioration of the rate of convergence of the outer iteration. The shifted symmetric part (4.1) has condition numbers often of the order of 10 or less, and is typically very easy to solve, at least in PDE problems. The solution of the shifted skew-symmetric part (4.2), on the other hand, is somewhat more problematic and warrants further research. Preliminary results show that when α is not too small, fairly accurate approximate solutions to the linear system (4.2) can be obtained in just 3–4 iterations of GMRES preconditioned

with an incomplete factorization. This inner-outer scheme, which requires using a flexible Krylov method (like FGMRES) as the outer iteration, is currently being investigated.

Acknowledgments. We gratefully acknowledge the many excellent suggestions from the anonymous referees, as well as helpful discussions with Howard Elman and Valeria Simoncini.

REFERENCES

[1] M. Arioli and G. Manzini, *A null space algorithm for mixed finite-element approximations of Darcy's equation*, Comm. Numer. Meth. Engrg., 18 (2002), pp. 645–657.

[2] 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.

[3] Z. Z. Bai, G. H. Golub, and J. Y. Pan, *Preconditioned Hermitian and Skew-Hermitian Splitting Methods for Non-Hermitian Positive Semidefinite Linear Systems*, Technical Report SCCM-02-12, Scientific Computing and Computational Mathematics Program, Department of Computer Science, Stanford University, Stanford, CA, 2002.

[4] 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.

[5] M. Benzi, *Solution of equality-constrained quadratic programming problems by a projection iterative method*, Rend. Mat. Appl., 13 (1993), pp. 275–296.

[6] M. Benzi, M. J. Gander, and G. H. Golub, *Optimization of the Hermitian and skew-Hermitian splitting iteration for saddle-point problems*, BIT, 43 (2003), pp. 881–900.

[7] 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.

[8] D. Bertaccini, G. H. Golub, S. Serra Capizzano, and C. Tablino Possio, *Preconditioned HSS Method for the Solution of Non-Hermitian Positive Definite Linear Systems*, Technical Report SCCM-02-11, Scientific Computing and Computational Mathematics Program, Department of Computer Science, Stanford University, Stanford, CA, 2002.

[9] J. T. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming*, SIAM, Philadelphia, 2001.

[10] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[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] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York, 1991.

[13] 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.

[14] P. Concus and G. H. 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, R. Glowinski and J. L. Lions, eds., Springer-Verlag, Berlin, 1976, pp. 56–65.

[15] E. de Sturler and J. Liesen, *Block-Diagonal Preconditioners for Indefinite Linear Algebraic Systems*, Report UIUCDCS-R-2002-2279, University of Illinois, Champaign, IL, 2002.

[16] I. S. Duff, N. I. M. Gould, J. K. Reid, J. A. Scott, and K. Turner, *The factorization of sparse symmetric indefinite matrices*, IMA J. Numer. Anal., 11 (1991), pp. 181–204.

[17] 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.

[18] N. Dyn and W. E. Ferguson, Jr., *The numerical solution of equality constrained quadratic programming problems*, Math. Comp., 41 (1983), pp. 165–170.

[19] S. C. Eisenstat, H. C. Elman, and M. H. Schultz, *Variational iterative methods for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal., 20 (1983), pp. 345–357.

[20] H. C. Elman, *Preconditioners for saddle point problems arising in computational fluid dynamics*, Appl. Numer. Math., 43 (2002), pp. 75–89.

[21] 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.

[22] 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. 665–688.
[23] 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, Lecture Notes in Math. 1457, Springer-Verlag, Berlin, 1990, pp. 28–43.
[24] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Application to the Numerical Solution of Boundary-Value Problems*, Stud. Math. Appl. 15, North-Holland, Amsterdam, 1983.
[25] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders, *Preconditioners for indefinite systems arising in optimization*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 292–311.
[26] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, New York, 1981.
[27] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, 1984.
[28] G. H. Golub and C. Greif, *On solving block-structured indefinite linear systems*, SIAM J. Sci. Comput., 24 (2003), pp. 2076–2092.
[29] G. H. Golub and D. Vanderstraeten, *On the preconditioning of matrices with skew-symmetric splittings*, Numer. Algorithms, 25 (2000), pp. 223–239.
[30] G. H. Golub and A. J. Wathen, *An iteration for indefinite systems and its application to the Navier–Stokes equations*, SIAM J. Sci. Comput., 19 (1998), pp. 530–539.
[31] G. H. Golub, X. Wu, and J.-Y. Yuan, *SOR-like methods for augmented systems*, BIT, 41 (2001), pp. 71–85.
[32] 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.
[33] E. Haber and U. M. Ascher, *Preconditioned all-at-once methods for large, sparse parameter estimation problems*, Inverse Problems, 17 (2001), pp. 1847–1864.
[34] E. Haber, U. M. Ascher, and D. Oldenburg, *On optimization techniques for solving nonlinear inverse problems*, Inverse Problems, 16 (2000), pp. 1263–1280.
[35] J. C. Haws, *Preconditioning KKT Systems*, Ph.D. thesis, Department of Mathematics, North Carolina State University, Raleigh, NC, 2002.
[36] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.
[37] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1991.
[38] M. Huhtanen, *A Hermitian Lanczos method for normal matrices*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 1092–1108.
[39] 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.
[40] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*, D. Van Nostrand Co., New York, 1960.
[41] A. Klawonn, *Block-triangular preconditioners for saddle point problems with a penalty term*, SIAM J. Sci. Comput., 19 (1998), pp. 172–184.
[42] J. Liesen, E. de Sturler, A. Sheffer, Y. Aydin, and C. Siefert, *Preconditioners for indefinite linear systems arising in surface parameterization*, in Proceedings of the 10th International Meshing Round Table, Sandia National Laboratories, eds., 2001, pp. 71–81.
[43] 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.
[44] J. Maryška, M. Rozložník, and M. Tůma, *Mixed-hybrid finite element approximation of the potential fluid flow problem*, J. Comput. Appl. Math., 63 (1995), pp. 383–392.
[45] J. Maryška, M. Rozložník, and M. Tůma, *Schur complement systems in the mixed-hybrid finite element approximation of the potential fluid flow problem*, SIAM J. Sci. Comput., 22 (2000), pp. 704–723.
[46] 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.
[47] W. Niethammer and R. S. Varga, *Relaxation methods for non-Hermitian linear systems*, Results Math., 16 (1989), pp. 308–320.
[48] 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.
[49] R. J. Plemmons, *A parallel block iterative scheme applied to computations in structural analysis*, SIAM J. Algebraic Discrete Methods, 7 (1986), pp. 337–347.

[50] W. Ren and J. Zhao, *Iterative methods with preconditioners for indefinite systems*, J. Comput. Math., 17 (1999), pp. 89–96.
[51] T. Rusten and R. Winther, *A preconditioned iterative method for saddlepoint problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 887–904.
[52] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.
[53] 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.
[54] V. Sarin and A. Sameh, *An efficient iterative method for the generalized Stokes problem*, SIAM J. Sci. Comput., 19 (1998), pp. 206–226.
[55] V. Simoncini and M. Benzi, *Spectral properties of the Hermitian and skew-Hermitian splitting preconditioner for saddle point problems*, SIAM J. Matrix Anal. Appl., to appear.
[56] G. Strang, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Wellesley, MA, 1986.
[57] J. C. Strikwerda, *An iterative method for solving finite difference approximations to the Stokes equations*, SIAM J. Numer. Anal., 21 (1984), pp. 447–458.
[58] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.
[59] E. L. Wachspress and G. J. Habetler, *An alternating-direction-implicit iteration technique*, J. Soc. Indust. Appl. Math., 8 (1960), pp. 403–424.
[60] J. S. Warsa, M. Benzi, T. A. Wareing, and J. E. Morel, *Preconditioning a Mixed Discontinuous Finite Element Method for Radiation Diffusion*, Technical Report LA-UR-01-4754, Los Alamos National Laboratory, 2001; Numer. Linear Algebra Appl., to appear.
[61] O. Widlund, *A Lanczos method for a class of nonsymmeric systems of linear equations*, SIAM J. Numer. Anal., 15 (1978), pp. 801–812.
[62] 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. 26, No. 1, pp. 42–69

JACOBI'S ALGORITHM ON COMPACT LIE ALGEBRAS*

M. KLEINSTEUBER†, U. HELMKE†, AND K. HÜPER‡

Abstract. A generalization of the cyclic Jacobi algorithm is proposed that works in an arbitrary compact Lie algebra. This allows, in particular, a unified treatment of Jacobi algorithms on different classes of matrices, e.g., skew-symmetric or skew-Hermitian Hamiltonian matrices. Wildberger has established global, linear convergence of the algorithm for the classical Jacobi method on compact Lie algebras. Here we prove local quadratic convergence for general cyclic Jacobi schemes.

Key words. Jacobi algorithm, compact Lie algebras, real root space decomposition, quadratic convergence, cost function, optimization

AMS subject classifications. 65F15, 17B20, 41A25, 74P20

DOI. 10.1137/S0895479802420069

1. Introduction.

The Jacobi algorithm for diagonalizing real symmetric or complex Hermitian matrices is a well-known eigenvalue method from numerical linear algebra [14]. The classical version of the algorithm has been first proposed by Jacobi (1846) [25], who successively applied Givens rotations that produce the largest decrease in the distance to diagonality. In contrast, modern approaches use cyclic sweep strategies to minimize the sum of squares of off-diagonal entries. Cyclic sweep strategies are more efficient than Jacobi's original approach, as one avoids the time consuming search for the largest off-diagonal element. Moreover, cyclic strategies are known to be well suited for parallel computing.

Variants of the Jacobi algorithm have been applied to various structured eigenvalue problems, including, e.g., the real skew-symmetric eigenvalue problem, [16, 24, 30], SVD computations [27], nonsymmetric eigenvalue problems [3, 6, 7, 34, 36], complex symmetric eigenproblems [8], and normal matrices [13]. For applications to different types of generalized eigenvalue problems, we refer to [2, 4, 15, 37]. For Jacobi methods applied to problems in systems theory, see [18, 19, 20].

The starting point for this paper is the Jacobi algorithm for the real skew-symmetric eigenvalue problem. For previous work in this direction, see [30] and, more recently, [16, 24] as well as the related papers [9, 28]. They all have in common that some kind of a block Jacobi method is used, i.e., multiparameter transformations that annihilate more than one pair of off-diagonal elements at the same time. In contrast, our approach exclusively uses one-parameter transformations.

Since the set of skew-symmetric matrices forms a Lie algebra it is not too surprising that the Jacobi algorithm can be extended to a general Lie algebraic setting. To our knowledge Wildberger [38] was the first who proposed a generalization of the classical Jacobi algorithm to arbitrary compact Lie algebras. The classification of compact Lie algebras shows that this approach essentially includes (i) the real skew-symmetric, (ii) the complex skew-Hermitian, (iii) the real skew-symmetric Hamiltonian, (iv) the complex skew-Hermitian Hamiltonian eigenvalue problem, and (v) some exceptional

*Received by the editors December 19, 2002; accepted for publication (in revised form) by P. Van Dooren September 17, 2003; published electronically August 6, 2004.
http://www.siam.org/journals/simax/26-1/42006.html
†Mathematical Institute, Würzburg University, Am Hubland, 97074 Würzburg, Germany (kleinsteuber@mathematik.uni-wuerzburg.de, helmke@mathematik.uni-wuerzburg.de).
‡National ICT Australia Ltd., Systems Engineering and Complex Systems (SEACS) Program, Locked Bag 8001, Canberra ACT 2601, Australia (knut.hueper@nicta.com.au).

cases. One might think that an algorithm for case (i) is also appropriate for case (iii), analogously for (ii) and (iv). However, to stay within the corresponding Lie algebra requires that the transformations be structure preserving, and therefore it is necessary to distinguish between these four cases. Nevertheless, following Wildberger, one can treat the above mentioned problems (i)–(v) on the same footing, meaning that the description and analysis of the Jacobi method can be carried out simultaneously for all four problems. This is exactly what is done in this paper, with an emphasis on establishing local quadratic convergence. There are several advantages of such an abstract approach. First, the theory is independent of any special coordinate representation of the underlying Lie algebra. This coordinate-free approach forces one to formulate the basic features of the algorithm in an abstract way, thus enabling one to work out the essential features of Jacobi algorithms. Moreover, the local convergence analysis for the numerical algorithm is in all these cases exactly the same. Our convergence analysis extends that described in the Ph.D. thesis by the third author [24], where elementary tools from global analysis were first used to prove local quadratic convergence for Jacobi-type methods. Questions of global convergence will not be discussed in this paper, albeit we expect that the ideas behind the proof of global convergence presented in [35] can be adopted. Instead, we restrict our discussion to local convergence properties.

The reader may have noticed that the real symmetric eigenvalue problem does not exactly fit into the framework developed in this paper, as the set of real symmetric matrices does not form a Lie algebra. In contrast, the Hermitian eigenvalue problem does. The reason is simply that the set of complex Hermitian matrices is up to multiplication with $\sqrt{-1}$ isomorphic to the compact Lie algebra of skew-Hermitian matrices. Of course, this process does not work for real symmetric matrices and therefore requires a different approach to that of this paper.

The general-purpose algorithm developed in this paper reduces to the skew-symmetric eigenvalue problem considered in [16] in the following way. Although the cited author uses block Jacobi methods to reduce the off-norm, it is possible to formulate an algorithm that uses only one-parameter rotations. Therefore the choice of the torus algebra is essential, because it determines the root spaces and hence the structure of the rotation matrices.

The paper is organized as follows. Basic definitions and results on Lie algebras appear in section 2. Furthermore, the structure of compact Lie algebras is analyzed and examples are given. In section 3 we discuss a cost function which can be regarded as the natural generalization of the familiar sum of squares function of off-diagonal entries. The critical points and the Hessian of this off-norm function are computed. The Jacobi algorithm on compact Lie algebras is formulated in section 4. Explicit formulas for the step size selections are given in section 5. The main result, namely the local quadratic convergence of the Jacobi algorithm, is presented in section 6. Finally, section 7 presents a pseudo code of the algorithm and section 8 includes some numerical experiments for the set of skew-Hermitian Hamiltonian matrices. The pseudo code is translated into a numerical algorithm for that case.

We like to emphasize that the notion of Hamiltonian matrices used in this paper follow the established convention in mathematics and especially Lie group theory. Thus a matrix is called Hamiltonian if it is skew-symmetric with respect to the standard symplectic form J. Some authors in linear algebra and systems theory differ from that definition by referring instead to a Hamiltonian matrix as one that is skew-Hermitian with respect to J. In the complex case this therefore leads to a different concept of Hamiltonian and associated symplectic tranformations. In particular, com-

plex Hamiltonian matrices in the latter sense do not define a complex Lie algebra, while this is true for the standard definition of Hamiltonian matrices.

2. Preliminaries on Lie algebras. The purpose of this section is to recall some basic facts and definitions about compact Lie algebras. For further information see, e.g., [1], [5], or [26]. In what follows, let $\mathbb{K}$ denote the fields $\mathbb{R}$, $\mathbb{C}$ of real or complex numbers, respectively.

DEFINITION 1. *A $\mathbb{K}$-vector space $\mathfrak{g}$ with a bilinear product*

$$[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \longrightarrow \mathfrak{g}$$

is called a Lie algebra *over $\mathbb{K}$ if*

(i) $[X,Y] = -[Y,X]$ *for all* $X, Y \in \mathfrak{g}$

(ii) $[[X,Y],Z] + [[Y,Z],X] + [[Z,X],Y] = 0$ (Jacobi identity).

Example 2. Let $\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$. Classical Lie algebras are given for example by

$$\begin{aligned}
\mathfrak{sl}(n,\mathbb{K}) &:= \{X \in \mathbb{K}^{n\times n} \mid \mathrm{tr}X = 0\},\\
\mathfrak{so}(n,\mathbb{K}) &:= \{X \in \mathbb{K}^{n\times n} \mid X^\top + X = 0\},\\
\mathfrak{sp}(n,\mathbb{K}) &:= \{X \in \mathbb{K}^{2n\times 2n} \mid X^\top J + JX = 0\},
\end{aligned}$$

where

$$J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$$

and I_n denotes the $(n \times n)$-identity matrix.

A Lie algebra $\mathfrak{g}$ over $\mathbb{R}$ ($\mathbb{C}$) is called *real* (*complex*). A *Lie subalgebra* $\mathfrak{h}$ is a subspace of $\mathfrak{g}$ for which $[\mathfrak{h},\mathfrak{h}] \subset \mathfrak{h}$ holds. In what follows, $\mathfrak{g}$ is always assumed to be a finite dimensional Lie algebra. For any $X \in \mathfrak{g}$, the *adjoint transformation* is the linear map

$$\mathrm{ad}_X : \mathfrak{g} \longrightarrow \mathfrak{g}, \qquad Y \longmapsto [X,Y] \tag{1}$$

and

$$\mathrm{ad} : \mathfrak{g} \longrightarrow \mathrm{End}(\mathfrak{g}), \qquad Y \longmapsto \mathrm{ad}_Y \tag{2}$$

is called the *adjoint representation* of $\mathfrak{g}$.

By means of (1) and (2), properties (i) and (ii) of Definition 1 are equivalent to $\mathrm{ad}_X Y = -\mathrm{ad}_Y X$ and $\mathrm{ad}_{[X,Y]} = \mathrm{ad}_X \mathrm{ad}_Y - \mathrm{ad}_Y \mathrm{ad}_X$, respectively. It follows immediately from property (i) that $\mathrm{ad}_X X = 0$ for all $X \in \mathfrak{g}$.

DEFINITION 3. *Let $\mathfrak{g}$ be a finite dimensional Lie algebra over $\mathbb{K}$. The symmetric bilinear form*

$$\kappa : \mathfrak{g} \times \mathfrak{g} \longrightarrow \mathbb{K}, \qquad \kappa(X,Y) \longmapsto \mathrm{tr}(\mathrm{ad}_X \circ \mathrm{ad}_Y) \tag{3}$$

is called the Killing form *of $\mathfrak{g}$.*

Let $\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$. Then

$$\begin{aligned}
\mathfrak{sl}(n,\mathbb{K}) &: \kappa(X,Y) = 2\,n\,\mathrm{tr}(XY) && \text{for } n \geq 2,\\
\mathfrak{so}(n,\mathbb{K}) &: \kappa(X,Y) = (n-2)\mathrm{tr}(XY) && \text{for } n \geq 3,\\
\mathfrak{sp}(n,\mathbb{K}) &: \kappa(X,Y) = 2(n+1)\mathrm{tr}(XY) && \text{for } n \geq 1;
\end{aligned}$$

cf. [17, p. 221], or [10, VI, 4]. Note that in [17], the notation $\mathfrak{sp}(2n, \mathbb{K})$ is used instead of $\mathfrak{sp}(n, \mathbb{K})$.

A Lie group is defined as a group together with a manifold structure such that the group operations are smooth functions. For an arbitrary Lie group G, the tangent space T_1G at the unit element $1 \in G$ possesses a Lie algebraic structure. This tangent space is called the Lie algebra of the Lie group G, denoted by $\mathfrak{g}$. The tangent mapping of the conjugation mapping in G at 1,

$$\mathrm{conj}_x(y) := xyx^{-1}$$

leads to the so-called *adjoint representation* of G

$$\mathrm{Ad} : G \times \mathfrak{g} \longrightarrow \mathfrak{g};$$

cf. [5, p. 2]. Considering now the tangent mapping of Ad with respect to g at 1 leads to the adjoint transformation (1). If G is a matrix group, then the elements of the corresponding Lie algebra can also be regarded as matrices; cf. [26, p. 53]. In this case the adjoint representation of $g \in G$ applied to $X \in \mathfrak{g}$ is given by

$$\mathrm{Ad}_g X = gXg^{-1},$$

i.e., by the usual similarity transformation of matrices, and the adjoint transformation is given by

$$\mathrm{ad}_Y X = YX - XY.$$

A basic property of the Killing form κ defined by (3) is its Ad-invariance, i.e.,

$$\kappa(\mathrm{Ad}_g X, \mathrm{Ad}_g Y) = \kappa(X, Y) \qquad \text{for all } X, Y \in \mathfrak{g}, g \in G. \tag{4a}$$

Differentiating the left side of this equation with respect to g gives

$$\mathrm{D}\kappa(\mathrm{Ad}_g X, \mathrm{Ad}_g Y) \cdot gZ = \kappa(\mathrm{Ad}_g(\mathrm{ad}_Z X), \mathrm{Ad}_g Y) + \kappa(\mathrm{Ad}_g X, \mathrm{Ad}_g(\mathrm{ad}_Z Y)),$$

where $gZ \in T_gG$ is in the tangent space of G at g. Therefore, using (4a) we get

$$\kappa(\mathrm{ad}_X Y, Z) = -\kappa(Y, \mathrm{ad}_X Z) \qquad \text{for all } X, Y, Z \in \mathfrak{g}. \tag{4b}$$

Definition 4. *A real finite dimensional Lie algebra* $\mathfrak{g}$ *is called* compact *if there exists a compact Lie group with Lie algebra* $\mathfrak{g}$.

Example 5. The following Lie algebras are compact (cf. [26, pp. 33, 36, and 66ff]):

$$\mathfrak{so}(n, \mathbb{R}) := \{S \in \mathbb{R}^{n \times n} \mid S^\top = -S\},$$
$$\mathfrak{u}(n, \mathbb{C}) := \{X \in \mathbb{C}^{n \times n} \mid X^* = -X\},$$
$$\mathfrak{su}(n, \mathbb{C}) := \{X \in \mathbb{C}^{n \times n} \mid X^* = -X, \mathrm{tr} X = 0\},$$
$$\mathfrak{sp}(n) := \mathfrak{u}(2n, \mathbb{C}) \cap \mathfrak{sp}(n, \mathbb{C}).$$

A finite dimensional Lie algebra $\mathfrak{g}$ admits a positive definite Ad-invariant bilinear form (cf. [26, p. 196, Proposition 4.24]). This property is used to show that the Killing form on compact Lie algebras is negative semidefinite (cf. [26, p. 197, Corollary 4.26]).

A Lie algebra $\mathfrak{g}$ is called *Abelian* if $[\mathfrak{g},\mathfrak{g}] = 0$. Let $\mathfrak{t} \subset \mathfrak{g}$ denote a maximal Abelian subalgebra of the compact Lie algebra $\mathfrak{g}$. Such a subalgebra is called *torus algebra* and the dual space is denoted as $\mathfrak{t}^*$. The maximal torus theorem (cf. [5, p. 152]) states that any two torus algebras, say $\mathfrak{t}, \mathfrak{t}'$, of a compact Lie algebra, are conjugate, i.e., there exists a $g \in G$, such that

$$\mathrm{Ad}_g \mathfrak{t} = \mathfrak{t}'.$$

Moreover, for a given $X \in \mathfrak{g}$ and a fixed torus algebra $\mathfrak{t}$ there exists $g \in G$ such that

$$\mathrm{Ad}_g X \in \mathfrak{t}.$$

The maximal torus theorem therefore generalizes the well-known fact that any skew-symmetric matrix is unitarily diagonalizable over $\mathbb{C}$.

To define the Jacobi algorithm one needs a set of optimizing directions in a compact Lie algebra. This is given by the *real root space decomposition*, which is an important tool in analyzing the structure of Lie algebras. For the remainder of this section, the root space decomposition of a compact Lie algebra is explained. Example 8 illustrates the correspondence between the root space decomposition and the off-diagonal entries of a skew-Hermitian matrix.

LEMMA 6. *Let $\mathfrak{g}$ be a compact Lie algebra and $X \in \mathfrak{g}$. Then*

$$\mathrm{ad}_X^* = -\mathrm{ad}_X \quad \textit{for all } X \in \mathfrak{g},$$

where adjoint $(\cdot)^*$ is defined relative to the Ad*-invariant inner product on $\mathfrak{g}$.*

Proof. Denote by B the Ad-invariant inner product on $\mathfrak{g}$; cf. [26, p. 197, Corollary 4.26]. Let $X, Y, Z \in \mathfrak{g}$. Then it holds that

$$B(\mathrm{ad}_X Y, Z) = B(Y, -\mathrm{ad}_X Z) = B(-\mathrm{ad}_X^* Y, Z). \qquad \square$$

Now fix a maximal Abelian subalgebra $\mathfrak{t} \subset \mathfrak{g}$. For $T_1, T_2 \in \mathfrak{t}$, it holds that $\mathrm{ad}_{T_1}\mathrm{ad}_{T_2} = \mathrm{ad}_{T_2}\mathrm{ad}_{T_1}$ and hence

$$\{\mathrm{ad}_T \mid T \in \mathfrak{t}\}$$

has a simultaneous eigenspace decomposition. Let X denote a simultaneous eigenvector of ad_T for all $T \in \mathfrak{t}$. By Lemma 6, ad_T possesses only purely imaginary eigenvalues and hence one has $\mathrm{ad}_T^2 X = -(\alpha(T))^2 X$, with $\alpha \in \mathfrak{t}^*$. To fix notation, a notion of positivity on $\mathfrak{t}^*$ is introduced. This can be done for example via lexicographic ordering; cf. [26, p. 109]. For $\alpha > 0$, we write

$$\mathfrak{g}_\alpha = \{X \in \mathfrak{g} \mid (\mathrm{ad}_T)^2 X = -(\alpha(T))^2 X \text{ for all } T \in \mathfrak{t}\}.$$

If $\mathfrak{g}_\alpha \neq 0$, we call $\mathfrak{g}_\alpha$ a *real root space* and α a *root*. The set of all positive roots is denoted by $\Sigma^+ \subset \mathfrak{t}^*$. Note that our notation slightly differs from that in the literature. For example, Duistermaat and Kolk [5] denote the real root spaces by $(\mathfrak{g}_\alpha \oplus \mathfrak{g}_{-\alpha}) \cap \mathfrak{g}$, where $\mathfrak{g}_\alpha, \mathfrak{g}_{-\alpha}$ are the complex root spaces of the complexification of $\mathfrak{g}$.

We summarize the above results.

PROPOSITION 7 (real root space decomposition). *Let $\mathfrak{g}$ be a compact Lie algebra and let Σ^+ denote the set of positive roots. Then $\mathfrak{g}$ decomposes orthogonally with respect to the Killing form into*

$$\mathfrak{g} = \mathfrak{t} \oplus \sum_{\alpha \in \Sigma^+} \mathfrak{g}_\alpha. \tag{5}$$

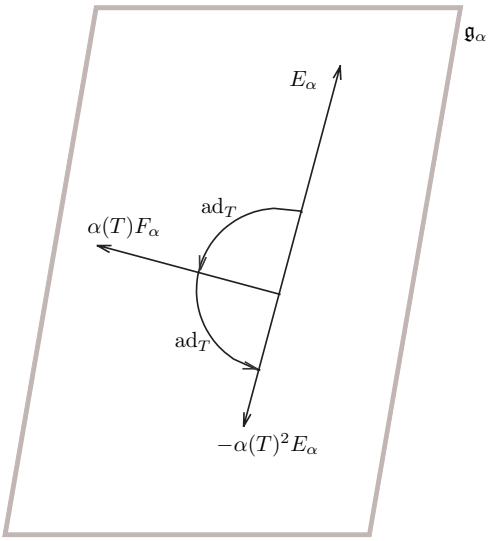


FIG. 2.1. *Action of* ad_T *in the real root space* $\mathfrak{g}_\alpha$ *with respect to an orthogonal basis* $\{E_\alpha, F_\alpha\}$.

Each real root space $\mathfrak{g}_\alpha$ *is of real dimension* 2. *It has an orthonormal basis* $\{E_\alpha, F_\alpha\}$ *with respect to* κ *such that for any* $T \in \mathfrak{t}$ *and* $\alpha \in \Sigma^+$ (*see Figure* 2.1)*:*

(i) $\mathrm{ad}_T E_\alpha = \alpha(T)F_\alpha$,
(ii) $\mathrm{ad}_T F_\alpha = -\alpha(T)E_\alpha$.

Proof. The first part is not proven and the reader is referred to [5, p. 146] and [26, p. 96, Proposition 2.21]. Let $E_\alpha \in \mathfrak{g}_\alpha$ arbitrary and let $T \in \mathfrak{t}$ with $\alpha(T) \neq 0$. Set

$$F_\alpha := \frac{1}{\alpha(T)}\mathrm{ad}_T E_\alpha.$$

Then $F_\alpha \neq 0$ since $\mathrm{ad}_T^2 E_\alpha = -\alpha(T)^2 E_\alpha \neq 0$ and $\kappa(F_\alpha, E_\alpha) = 0$ because

$$\kappa(\mathrm{ad}_T E_\alpha, E_\alpha) = \kappa(E_\alpha, -\mathrm{ad}_T E_\alpha) = -\kappa(\mathrm{ad}_T E_\alpha, E_\alpha). \qquad \square$$

In what follows, the following notation will be convenient. For $a, b \in \mathbb{R}$ and $X = aE_\alpha + bF_\alpha \in \mathfrak{g}_\alpha$ define

$$\overline{X} := -bE_\alpha + aF_\alpha. \tag{6}$$

Example 8. Let $\mathrm{i} := \sqrt{-1}$. Let $\mathfrak{g} = \mathfrak{su}(3, \mathbb{C})$ and fix a torus algebra $\mathfrak{t}$ by

$$\mathfrak{t} = \left\{ \begin{bmatrix} \mathrm{i}x_1 & 0 & 0 \\ 0 & \mathrm{i}x_2 & 0 \\ 0 & 0 & \mathrm{i}x_3 \end{bmatrix} \;\middle|\; x_1, x_2, x_3 \in \mathbb{R},\ \sum_{k=1}^{3} x_k = 0 \right\}.$$

Then the real root spaces and the corresponding roots turn out to be

$$\mathfrak{g}_{\alpha_1} = \left\{ \begin{bmatrix} 0 & \lambda+\mathrm{i}\nu & 0 \\ -\lambda+\mathrm{i}\nu & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \;\middle|\; \lambda,\nu \in \mathbb{R} \right\},\ \alpha_1 \left(\begin{bmatrix} \mathrm{i}x_1 & 0 & 0 \\ 0 & \mathrm{i}x_2 & 0 \\ 0 & 0 & \mathrm{i}x_3 \end{bmatrix} \right) = x_1 - x_2,$$

$$\mathfrak{g}_{\alpha_2} = \left\{ \begin{bmatrix} 0 & 0 & \lambda+\mathrm{i}\nu \\ 0 & 0 & 0 \\ -\lambda+\mathrm{i}\nu & 0 & 0 \end{bmatrix} \;\middle|\; \lambda,\nu \in \mathbb{R} \right\},\ \alpha_2 \left(\begin{bmatrix} \mathrm{i}x_1 & 0 & 0 \\ 0 & \mathrm{i}x_2 & 0 \\ 0 & 0 & \mathrm{i}x_3 \end{bmatrix} \right) = x_1 - x_3,$$

$$\mathfrak{g}_{\alpha_3} = \left\{ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \lambda+\mathrm{i}\nu \\ 0 & -\lambda+\mathrm{i}\nu & 0 \end{bmatrix} \;\middle|\; \lambda,\nu \in \mathbb{R} \right\},\ \alpha_3 \left(\begin{bmatrix} \mathrm{i}x_1 & 0 & 0 \\ 0 & \mathrm{i}x_2 & 0 \\ 0 & 0 & \mathrm{i}x_3 \end{bmatrix} \right) = x_2 - x_3. \quad \square$$

The way real root spaces of a compact Lie algebra are related to each other is similar to the way complex root spaces of a complex semisimple Lie algebra [26, p. 96] are related. We write $\alpha > \beta$ if $\alpha - \beta$ is positive.

LEMMA 9. *Let $\mathfrak{g}_\alpha$ and $\mathfrak{g}_\beta$ be real root spaces of the compact Lie algebra $\mathfrak{g}$. Without loss of generality assume $\alpha > \beta$. Then*

$$[\mathfrak{g}_\alpha, \mathfrak{g}_\beta] = \mathfrak{g}_{\alpha+\beta} \oplus \mathfrak{g}_{\alpha-\beta}$$

holds, where

$$\mathfrak{g}_{\alpha+\beta} := 0 \qquad \textit{if} \quad \alpha+\beta \notin \Sigma^+,$$
$$\mathfrak{g}_{\alpha-\beta} := 0 \qquad \textit{if} \quad \alpha-\beta \notin \Sigma^+.$$

Proof. Direct consequence of the definition of real root spaces [5, p. 146] and the relations between complex root spaces [26, p. 88, Proposition 2.5]. $\square$

We need the following lemmata for further calculation.

LEMMA 10. *Let $\{E_\gamma, F_\gamma\}$ be a basis of the real root space $\mathfrak{g}_\gamma$ as in Proposition 7. Then $T_\gamma := [E_\gamma, F_\gamma]$ lies in the maximal torus algebra $\mathfrak{t}$ and moreover, $\gamma(T_\gamma) > 0$.*

Proof. By the Jacobi identity and Proposition 7 for an arbitrary $H \in \mathfrak{t}$ it holds that

$$\mathrm{ad}_H [E_\gamma, F_\gamma] = 0.$$

Hence, $[E_\gamma, F_\gamma] \in \mathfrak{t}$. Now let $X = x\, E_\gamma + y\, F_\gamma$ with $(x,y) \in \mathbb{R}^2 - \{0\}$ and let B be a positive definite bilinear Ad-invariant form on $\mathfrak{g}$; cf. [26, p. 196]. Moreover (cf. (6))

$$\begin{aligned}
\gamma([E_\gamma, F_\gamma]) B\Big(\overline{X}, \overline{X}\Big) &= B\Big(\overline{X}, \mathrm{ad}_{[E_\gamma, F_\gamma]} X\Big) \\
&= -y\, B\Big(E_\gamma, x\, \mathrm{ad}_{E_\gamma}\mathrm{ad}_{F_\gamma} E_\gamma - y\, \mathrm{ad}_{F_\gamma}\mathrm{ad}_{E_\gamma} F_\gamma\Big) \\
&\quad + x\, B\Big(F_\gamma, x\, \mathrm{ad}_{E_\gamma}\mathrm{ad}_{F_\gamma} E_\gamma - y\, \mathrm{ad}_{F_\gamma}\mathrm{ad}_{E_\gamma} F_\gamma\Big) \\
&= -y\, B\Big(E_\gamma, -y\, \mathrm{ad}_{F_\gamma}\mathrm{ad}_{E_\gamma} F_\gamma\Big) + x\, B\Big(F_\gamma, x\, \mathrm{ad}_{E_\gamma}\mathrm{ad}_{F_\gamma} E_\gamma\Big) \\
&= y^2\, B\Big(\mathrm{ad}_{E_\gamma} F_\gamma, \mathrm{ad}_{E_\gamma} F_\gamma\Big) + x^2\, B\Big(\mathrm{ad}_{E_\gamma} F_\gamma, \mathrm{ad}_{E_\gamma} F_\gamma\Big) > 0,
\end{aligned}$$

since $[E_\gamma, F_\gamma] \neq 0$. $\square$

LEMMA 11. *For arbitrary $H \in \mathfrak{t}$ and any* Ad*-invariant bilinear form $(\cdot,\cdot)$ it holds that*

$$(H, T_\gamma) = \frac{\gamma(H)}{\gamma(T_\gamma)}(T_\gamma, T_\gamma) \qquad \text{for all } \gamma \in \Sigma^+.$$

Proof. We use the definition of T_γ, cf. Lemma 10. Let $H \in \mathfrak{t}$. Then

$$\begin{aligned}\gamma(T_\gamma)(H, \mathrm{ad}_{E_\gamma} F_\gamma) &= \gamma(T_\gamma)(\mathrm{ad}_H E_\gamma, F_\gamma)\\ &= \gamma(H)\gamma(T_\gamma)(F_\gamma, F_\gamma)\\ &= \gamma(H)(\mathrm{ad}_{[E_\gamma,F_\gamma]}E_\gamma\,,\, F_\gamma)\\ &= \gamma(H)(-\mathrm{ad}_{E_\gamma}\mathrm{ad}_{E_\gamma}F_\gamma\,,\, F_\gamma)\\ &= \gamma(H)(T_\gamma, T_\gamma). \qquad \square\end{aligned}$$

3. A cost function. Since Jacobi-type methods can be considered as optimization algorithms [22, 23], it is instrumental to make a thorough analysis of the cost function we want to minimize. For the Jacobi algorithm, one considers the so-called off-norm function of a square matrix $X = (x_{ij})$, defined as the sum of squares of all its off-diagonal elements

$$\mathrm{off} : \mathbb{R}^{n\times n} \longrightarrow [0,\infty), \qquad \mathrm{off}(X) = \sum_{i\neq j} x_{ij}^2.$$

In this section, a generalization of the off-norm of matrices is discussed. The set of critical points is computed as well as the Hessian. These calculations are essential steps towards the analysis of the local convergence properties of our algorithm.

Let G be a compact Lie group with compact Lie algebra $\mathfrak{g}$ and real root space decomposition (5). Let κ denote the Killing form on $\mathfrak{g}$. Denote by

$$\mathrm{p} : \mathfrak{g} \longrightarrow \mathfrak{t} \tag{7}$$

the orthogonal projection on $\mathfrak{t}$ with respect to κ. Any $X \in \mathfrak{g}$ decomposes into

$$X = X_0 + \sum_{\alpha\in\Sigma^+} X_\alpha$$

corresponding to (5), with $X_0 := \mathrm{p}(X)$. For a given $S \in \mathfrak{g}$ let

$$\mathcal{O}_S := \{\mathrm{Ad}_g S \mid g \in G\}$$

denote the *adjoint orbit of* S. A cost function is defined as

$$f : \mathcal{O}_S \longrightarrow [0,\infty), \qquad X \longmapsto -\kappa(X - X_0, X - X_0). \tag{8a}$$

By negative semidefiniteness of κ on $\mathfrak{g}$, f is nonnegative. By orthogonality of the root space decomposition (5), it holds that

$$f(X) = -\kappa(X, X) + \kappa(X_0, X_0)$$

and $\kappa(X, X) = \kappa(S, S)$ is constant along the orbit $\mathcal{O}_S$, cf. (4a). Moreover, by Proposition 7 and Lemma 12, (ii), the cost function defined by (8a) is equal to

$$f(X) = -\kappa(S, S) - 2\sum_{\alpha\in\Sigma^+} \alpha^2(X_0). \tag{8b}$$

This shows that f is the natural generalization of the off-norm function on a compact Lie algebra.

We now analyze the cost function (8a) in detail. The following result summarizes two properties that will be needed for subsequent calculations. Recall that $X_0 = \mathrm{p}(X)$ by definition.

LEMMA 12. *Let $\mathfrak{g}$ be a compact Lie algebra with fixed maximal torus algebra $\mathfrak{t} \subset \mathfrak{g}$. Let* p *be as in* (7) *and $X, Y \in \mathfrak{g}$. Then the following holds:*

(i) $\mathrm{p}(\mathrm{ad}_Y X_0) = 0$,

(ii) $\kappa(Y_0, X - X_0) = 0$.

Proof. By linearity of the adjoint transformation (4b) and by the root space decomposition (5) of $\mathfrak{g}$ it holds that

$$\mathrm{ad}_Y X_0 = \mathrm{ad}_{Y_0} X_0 + \sum_{\alpha \in \Sigma^+} \mathrm{ad}_{Y_\alpha} X_0.$$

The summand $\mathrm{ad}_{Y_\alpha} X_0$ lies in $\mathfrak{g}_\alpha$ for all $\alpha \in \Sigma^+$ and $\mathrm{ad}_{Y_0} X_0 = 0$ holds. Hence $\mathrm{ad}_Y X_0$ has no $\mathfrak{t}$-component and therefore

$$\mathrm{p}(\mathrm{ad}_Y X_0) = 0.$$

Statement (ii) is a direct consequence of (7). □

THEOREM 13. *Let κ be the Killing form on the compact Lie algebra $\mathfrak{g}$, $S \in \mathfrak{g}$ arbitrary and* p *as in* (7). *Let*

$$f : \mathcal{O}_S \longrightarrow [0, \infty), \qquad X \longmapsto -\kappa\Big(X - \mathrm{p}(X), X - \mathrm{p}(X)\Big)$$

as above.

(a) *The following statements are equivalent:*

(i) *$X \in \mathcal{O}_S$ is a critical point of f,*

(ii) $\mathrm{ad}_{X_0} X = 0$,

(iii) $\alpha(X_0)\ X_\alpha = 0$ *for all* $\alpha \in \Sigma^+$.

(b) *Let Z be a critical point of f and let $\mathrm{ad}_H Z \in T_Z \mathcal{O}_S$ be an arbitrary element of the tangent space at Z. Then the Hessian of f at Z is*

$$\begin{aligned} \mathsf{H}_f(Z) : \quad T_Z\mathcal{O}_S \times T_Z\mathcal{O}_S \quad &\longrightarrow \quad \mathbb{R}, \\ (\mathrm{ad}_H Z, \mathrm{ad}_H Z) \quad &\longmapsto -2\kappa(\mathrm{ad}_H Z, \mathrm{ad}_H Z_0 - \mathrm{p}(\mathrm{ad}_H Z)). \end{aligned}$$

Proof. (a) For arbitrary $H, X \in \mathfrak{g}$ let $\gamma : \mathbb{R} \longrightarrow \mathcal{O}_S$, $\gamma(t) = \mathrm{Ad}_{\exp(tH)} X$, be a smooth curve through X. The derivative of the cost function (8a) at X can be calculated in the following way. By Lemma 12 and the Ad-invariance of the Killing form κ (4b) we have

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\Big(f \circ \gamma\Big)(t)\Big|_{t=0} &= -2\kappa(\mathrm{ad}_H X - \mathrm{p}(\mathrm{ad}_H X), X - X_0) \\ &= -2\kappa(-\mathrm{ad}_X H + \mathrm{p}(\mathrm{ad}_X H), X - X_0) \\ &= -2\kappa(H, \mathrm{ad}_X(X - X_0)) \\ &= -2\kappa(H, \mathrm{ad}_{X_0} X). \end{aligned}$$

Hence

$$Df(X) \equiv 0 \quad \Longleftrightarrow \quad \mathrm{ad}_{X_0} X \in \mathrm{rad}_\kappa,$$

where rad_κ denotes the radical of the Killing form κ.

On compact Lie algebras, the radical rad_κ coincides with the center of $\mathfrak{g}$ [5, p. 148] and therefore $\mathrm{ad}_{X_0} X$ has to coincide with its projection on $\mathfrak{t}$. Using Lemma 12 again, one obtains $\mathrm{ad}_{X_0} X = 0$. Hence for a critical point X it holds that

$$\sum_{\alpha\in\Sigma^+} \mathrm{ad}_{X_0} X_\alpha = 0$$
$$\Longleftrightarrow$$
$$\mathrm{ad}_{X_0} X_\alpha = 0 \quad \text{for all } \alpha \in \Sigma^+$$
$$\Longleftrightarrow$$
$$\alpha(X_0)\cdot X_\alpha = 0 \quad \text{for all } \alpha \in \Sigma^+.$$

(b) By a simple but lengthy computation, for arbitrary $H \in \mathfrak{g}$ and $\gamma(t) := \mathrm{Ad}_{\exp(tH)} Z$, it follows that

$$\begin{aligned}
&\frac{\mathrm{d}^2}{\mathrm{d}t^2}(f\circ\gamma)(t)\Big|_{t=0} \\
&= -\frac{\mathrm{d}^2}{\mathrm{d}t^2}\kappa(\mathrm{Ad}_{\exp(tH)}Z - \mathrm{p}(\mathrm{Ad}_{\exp(tH)}Z), \mathrm{Ad}_{\exp(tH)}Z - \mathrm{p}(\mathrm{Ad}_{\exp(tH)}Z))\Big|_{t=0} \\
&= -2\frac{\mathrm{d}}{\mathrm{d}t}\kappa(\mathrm{ad}_H \mathrm{Ad}_{\exp(tH)}Z - \mathrm{p}(\mathrm{ad}_H \mathrm{Ad}_{\exp(tH)}Z), \mathrm{Ad}_{\exp(tH)}Z - \mathrm{p}(\mathrm{Ad}_{\exp(tH)}Z))\Big|_{t=0} \\
&= -2\kappa(\mathrm{ad}_H^2 Z - \mathrm{p}(\mathrm{ad}_H^2 Z), Z - Z_0) - 2\kappa(\mathrm{ad}_H Z - \mathrm{p}(\mathrm{ad}_H Z), \mathrm{ad}_H Z - \mathrm{p}(\mathrm{ad}_H Z)) \\
&= -2\kappa(\mathrm{ad}_H Z, -\mathrm{ad}_H Z + \mathrm{ad}_H Z_0) - 2\kappa(\mathrm{ad}_H Z, \mathrm{ad}_H Z - \mathrm{p}(\mathrm{ad}_H Z)) \\
&= -2\kappa(\mathrm{ad}_H Z, \mathrm{ad}_H Z_0 - \mathrm{p}(\mathrm{ad}_H Z)). \qquad \square
\end{aligned}$$

Note that for any $\xi \in T_Z\mathcal{O}_S$ in the tangent space at a critical point Z, elements $H \in \mathfrak{g}$ satisfying $\xi = \mathrm{ad}_H Z$ are not uniquely determined. That is $\xi = \mathrm{ad}_H Z = \mathrm{ad}_{H+C} Z$ whenever $[C, Z] = 0$. Nevertheless,

$$\kappa(\mathrm{ad}_{H+C} Z, \mathrm{ad}_{H+C} Z_0 - \mathrm{p}(\mathrm{ad}_{H+C} Z)) = \kappa(\mathrm{ad}_H Z, \mathrm{ad}_H Z_0 - \mathrm{p}(\mathrm{ad}_H Z))$$

holds. Thus the selection of elements H with $\xi = \mathrm{ad}_H Z$ does not affect the validity of the expression for the Hessian.

The next two lemmata contain information about the restriction of the Hessian to one dimensional subspaces of $T_Z\mathcal{O}_S$. It turns out that, whenever the critical point Z is not a global minimum, there exists a one dimensional subspace of $T_Z\mathcal{O}_S$ on which the restriction of the Hessian is negative definite. Hence the cost function possesses only global minima. A similar argument shows that the local maxima of the cost function are global. One concludes that all other critical points are saddle points.

LEMMA 14. *Let $\beta \in \Sigma^+$ be a real root, $\Omega \neq 0$ an arbitrary element of the real root space $\mathfrak{g}_\beta$ and let $Z \in \mathcal{O}_S$ denote a critical point of the cost function* (8a). *Let $Z_0 \in \mathfrak{t}$ denote the torus algebra component of Z. Then*

$$\beta(Z_0) \neq 0 \qquad \textit{implies} \qquad \mathsf{H}_f(Z)(\mathrm{ad}_\Omega Z, \mathrm{ad}_\Omega Z) > 0.$$

Proof. Let $\beta(Z_0) \neq 0$. Then $Z_\beta = 0$ by Theorem 13. As $\mathrm{ad}_\Omega Z_\alpha$ has no torus algebra component for $\alpha \neq \beta$, one obtains

$$\mathrm{p}(\mathrm{ad}_\Omega Z_\alpha) = 0 \qquad \text{for all } \alpha \in \Sigma^+.$$

Moreover, $\mathrm{ad}_\Omega Z_\alpha$ lies for any $\alpha \in \Sigma^+$ in the orthogonal complement of $\mathfrak{g}_\beta$ (cf. Lemma 9) and therefore

$$\kappa\left(\sum_{\alpha\in\Sigma^+} \mathrm{ad}_\Omega Z_\alpha, \mathrm{ad}_\Omega Z_0\right) = -\kappa\left(\sum_{\alpha\in\Sigma^+} \mathrm{ad}_\Omega Z_\alpha, \mathrm{ad}_{Z_0}\Omega\right) = 0$$

also holds. We compute the restriction of the Hessian evaluated at Z to the subspace $\mathbb{R}\cdot\mathrm{ad}_\Omega Z$.

$$\begin{aligned}\frac{1}{2}\mathsf{H}_f(Z)(\mathrm{ad}_\Omega Z, \mathrm{ad}_\Omega Z) &= -\kappa(\mathrm{ad}_\Omega Z_0, \mathrm{ad}_\Omega Z_0) - \kappa\left(\sum_{\alpha\in\Sigma^+} \mathrm{ad}_\Omega Z_\alpha, \mathrm{ad}_\Omega Z_0\right)\\ &= -\kappa(\mathrm{ad}_\Omega Z_0, \mathrm{ad}_\Omega Z_0)\\ &= -\beta(Z_0)^2\kappa(\Omega,\Omega) > 0. \qquad \square\end{aligned}$$

LEMMA 15. *Let $\gamma \in \Sigma^+$ be a real root, let $\{\Omega_1, \Omega_2\}$ be some basis of $\mathfrak{g}_\gamma$, and let $Z \in \mathcal{O}_S$ be a critical point of the cost function defined by* (8a). *Then*

$$Z_\gamma \neq 0 \qquad \textit{implies} \qquad \mathsf{H}_f(Z)(\mathrm{ad}_{\Omega_j} Z, \mathrm{ad}_{\Omega_j} Z) < 0$$

for either $j = 1$ or $j = 2$.

Proof. Let $\Omega \in \{\Omega_1, \Omega_2\}$ such that $\Omega \notin \mathbb{R}\cdot Z_\gamma$. By Theorem 13, $\gamma(Z_0) = 0$ and therefore (cf. (6))

$$\mathrm{ad}_\Omega Z_0 = -\mathrm{ad}_{Z_0}\Omega = \pm\gamma(Z_0)\overline{\Omega} = 0.$$

The Hessian restricted to the subspace $\mathbb{R}\cdot\mathrm{ad}_\Omega Z$ is

$$\begin{aligned}\frac{1}{2}\mathsf{H}_f(Z)(\mathrm{ad}_\Omega Z, \mathrm{ad}_\Omega Z) &= \kappa\left(\sum_{\alpha\in\Sigma^+} \mathrm{ad}_\Omega Z_\alpha, \sum_{\alpha\in\Sigma^+} \mathrm{p}(\mathrm{ad}_\Omega Z_\alpha)\right)\\ &= \kappa\left(\sum_{\alpha\in\Sigma^+} \mathrm{p}(\mathrm{ad}_\Omega Z_\alpha), \sum_{\alpha\in\Sigma^+} \mathrm{p}(\mathrm{ad}_\Omega Z_\alpha)\right)\\ &= \kappa(\mathrm{ad}_\Omega Z_\gamma, \mathrm{ad}_\Omega Z_\gamma) < 0;\end{aligned}$$

cf. Lemma 10. $\square$

As a consequence of the last two lemmata we obtain the following.

PROPOSITION 16.

(i) *The local minima of the cost function* (8a) *are global minima. The set of the minima is $\mathcal{O}_S \cap \mathfrak{t}$.*

(ii) *The local maxima of the cost function* (8a) *are global maxima. The set of the maxima is $\mathcal{O}_S \cap \mathfrak{t}^\perp$ where $\mathfrak{t}^\perp$ denotes the orthogonal complement of $\mathfrak{t}$ with respect to κ.*

Proof. (i) Let Z be a local minimum of (8a); then

$$\mathsf{H}_f(Z)(\mathrm{ad}_\Omega Z, \mathrm{ad}_\Omega Z) \geq 0 \quad \text{for all } \Omega \in \mathfrak{g}.$$

By Lemma 15,

$$Z_\gamma = 0 \quad \text{for all } \gamma \in \Sigma^+.$$

Hence $\mathrm{p}(Z) = Z$ and $f(Z) = 0$.

(ii) Now let Z be a local maximum of (8a); then

$$\mathsf{H}_f(Z)(\mathrm{ad}_\Omega Z, \mathrm{ad}_\Omega Z) \leq 0 \quad \text{for all } \Omega \in \mathfrak{g}.$$

By Lemma 14,

$$\gamma(Z_0) = 0 \quad \text{for all } \gamma \in \Sigma^+. \tag{9}$$

By comparing (9) with (8b) it follows that $f(Z) = -\kappa(S,S)$ and therefore Z is a global maximum of f. It follows immediately from (9) that $Z_0 \in \mathfrak{z}_\mathfrak{g}$, the center of $\mathfrak{g}$, and hence $Z \in \mathfrak{t}^\perp$. Note that $\mathfrak{t} \cap \mathfrak{t}^\perp = \mathfrak{z}_\mathfrak{g}$. □

The next theorem characterizes the critical points of the cost function (8a).

THEOREM 17.

(a) *A critical point Z of* (8a) *is a saddle point if and only if*

$$0 < f(Z) < -\kappa(S,S).$$

(b) *The set of minima of the cost function* (8a) *is finite.*

Proof. (a) Direct consequence of Proposition 16.

(b) The set of minima of the cost function (8a) is exactly the intersection of $\mathcal{O}_S$ with the torus algebra $\mathfrak{t}$. By Weyl's covering theorem [5, p. 153, section 3.7], we conclude that $|\mathcal{O}_S \cap \mathfrak{t}|$ is finite. □

4. The algorithm. As mentioned in the introduction, a Lie algebraic version of the classical Jacobi algorithm has already been published by Wildberger; cf. [38]. Proceeding from the real root space decomposition (5) of a compact Lie algebra $\mathfrak{g}$, Wildberger decomposes a given $Z \in \mathfrak{g}$ into torus algebra and root space components, i.e., $Z = Z_0 + \sum_\alpha Z_\alpha$. He shows the existence of a sequence of Lie algebra elements $(Z^{(1)}, Z^{(2)}, \dots)$, for which the following holds.

(i) $Z^{(k+1)} = \mathrm{Ad}_{g_k} Z^{(k)}$, where g_k only depends on $Z_\alpha^{(k)}$ and α is chosen such that $||Z_\alpha^{(k)}|| = \max_{\gamma \in \Sigma^+} ||Z_\gamma^{(k)}||$,

(ii) $Z^{(k+1)}$ has no $\mathfrak{g}_\alpha$ component,

(iii) the sequence $(Z^{(k)})$ converges to some torus algebra element.

The method described in [38] uses only $SU(2,\mathbb{C})$ transformations in a noncyclic manner which is completely analogous to Jacobi's original approach based on orthogonal transformations to annihilate the off-diagonal elements having greatest modulus; cf. [25].

We extend this construction by formulating in full generality a cyclic Jacobi algorithm on compact Lie algebras. The algorithm proceeds as follows. Let $G_1, \dots, G_M$ be closed one-parameter subgroups of the compact Lie group G. Then in a first step we minimize the restriction of the cost function (8a) to the orbit of the initial point $Z \in \mathfrak{g}$ under the adjoint action of G_1. Let $Z^{(1)} \in \mathrm{Ad}_{G_1} Z$ denote that minimum. The next step is done by minimizing the restriction of (8a) to $\mathrm{Ad}_{G_2} Z^{(1)}$ and so on until arriving at $Z^{(M)}$; cf. Figure 4.1. This procedure is called a *sweep*, and iterating sweeps forms the algorithm.

More precisely, let N denote the number of real root spaces of $\mathfrak{g}$ and choose

$$\mathfrak{B} = \{\Omega_1, \dots, \Omega_{2N}\} \tag{10}$$

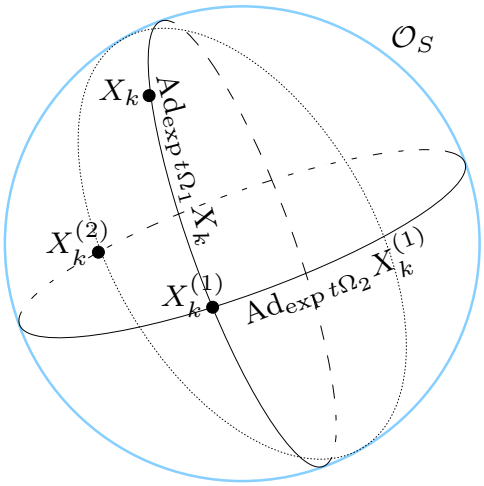


FIG. 4.1. *Illustration of the first step of the cyclic Jacobi sweep.*

as a basis of $\mathfrak{g}/\mathfrak{t}$, where for $i = 1, \ldots, N$, the set $\{\Omega_{2i-1}, \Omega_{2i}\}$ denotes orthogonal basis vectors of the real root space $\mathfrak{g}_{\alpha_i}$; cf. Proposition 7. For $\Omega \in \mathfrak{B}$ consider

$$r_\Omega : \mathcal{O}_S \times \mathbb{R} \longrightarrow \mathcal{O}_S, \qquad r_\Omega(X,t) := \mathrm{Ad}_{\exp(t\Omega)} X.$$

Furthermore, *step size selections*, depending on $\Omega_i \in \mathfrak{B}, i = 1, \ldots, 2N$, are defined as

$$t_*^{(i)} : \mathcal{O}_S \longrightarrow \mathbb{R},$$

$$t_*^{(i)}(X) := \begin{cases} 0, & \text{if } \big(f \circ \mathrm{Ad}_{\exp(t\Omega_i)}\big)(X) = f(X) \quad \text{for all } t \in \mathbb{R} \\ \arg\min_{t\in\mathbb{R}} \big(f \circ \mathrm{Ad}_{\exp(t\Omega_i)}\big)(X) & \text{otherwise.} \end{cases} \tag{11}$$

To guarantee uniqueness, $\arg\min_{t\in\mathbb{R}} \big(f \circ \mathrm{Ad}_{\exp(t\Omega_i)}\big)(X)$ denotes that $t \in \mathbb{R}$ being of smallest absolute value. In case there are two such minimal values $\pm t$, we choose the positive solution $t > 0$. A more explicit formula for (11) is given in section 5. Sweeps are defined as follows.

Cyclic Jacobi Sweep

$$\begin{aligned} X_k^{(1)} &:= r_{\Omega_1}\left(X_k^{(0)}, t_*^{(1)}\left(X_k^{(0)}\right)\right), \\ X_k^{(2)} &:= r_{\Omega_2}\left(X_k^{(1)}, t_*^{(2)}\left(X_k^{(1)}\right)\right), \\ &\vdots \\ X_k^{(2N)} &:= r_{\Omega_{2N}}\left(X_k^{(2N-1)}, t_*^{(2N)}\left(X_k^{(2N-1)}\right)\right). \end{aligned} \tag{12}$$

The Jacobi algorithm consists of iterating sweeps.

JACOBI'S ALGORITHM.

1. Let $X_0, X_1, \ldots, X_k \in \mathcal{O}_S$ be given for some $k \in \mathbb{N}$.
2. Set $X_k^{(0)} := X_k$ and define the sequence $X_k^{(1)}, \ldots, X_k^{(2N)}$ as in (12). (13)
3. Set $X_{k+1} := X_k^{(2N)}$ and continue with the next sweep.

Note that the smallest Lie algebra containing a root space is isomorphic to $\mathfrak{su}(2,\mathbb{C})$, see proof of Proposition 18. Therefore the Lie algebra $\mathfrak{g}$ is the (nondirect) sum

$$\mathfrak{g} = \sum_N \mathfrak{su}(2,\mathbb{C}) + \mathfrak{z}_\mathfrak{g},$$

where $\mathfrak{z}_\mathfrak{g}$ denotes the center of $\mathfrak{g}$. Hence sweep operations can in principle be organized via $SU(2,\mathbb{C})$ suboperations minimizing simultaneously along the directions $\{\Omega_{2i-1}, \Omega_{2i}\}$ for $i = 1, \ldots, N$ as has been done by Wildberger [38]. Such an approach leads to so-called block Jacobi methods as in each step the cost function restricted to a *two*-dimensional subset is minimized (cf. [22, 23]); see also [28] for minimization over higher dimensional subsets. Although in this paper we do not follow this idea and restrict ourselves to the algorithm as described above, i.e., minimizing along one dimensional subsets, our algorithm can also be considered as minimizing along the directions $\{\Omega_{2i-1}, \Omega_{2i}\}$ for $i = 1, \ldots, N$ simultaneously, as these two processes do not influence each other; cf. Proposition 19 and the remark that follows.

Torus algebra directions $T \in \mathfrak{t}$ can be omitted from the minimization process as the cost function (8b) is constant along the orbits of the generated one-parameter groups, i.e.,

$$\mathrm{p}\Big(\mathrm{Ad}_{\exp(tT)}X\Big) = \mathrm{p}(X)$$

holds for all $T \in \mathfrak{t}$ and $t \in \mathbb{R}$, because Lemma 12 implies

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{p}\Big(\mathrm{Ad}_{\exp(tT)}X\Big) = \mathrm{p}\Big(\mathrm{ad}_T\mathrm{Ad}_{\exp(tT)}X\Big) = 0.$$

PROPOSITION 18. *Let $X_\alpha \in \mathfrak{g}_\alpha$ and $Y \in \mathfrak{g}$ arbitrary. Then*

(i) $\mathrm{Ad}_{\exp(\mathbb{R}\cdot X_\alpha)}Y \cong S^1$ *if* $[Y, X_\alpha] \neq 0$ *and*

(ii) $\mathrm{Ad}_{\exp(\mathbb{R}\cdot X_\alpha)}Y \cong \{1\}$ *if* $[Y, X_\alpha] = 0$.

Thus the cost function restricted to $\mathrm{Ad}_{\exp(\mathbb{R}\cdot X_\alpha)}Y$ possesses at least one minimum and Algorithms (12) *and* (13) *are therefore well defined.*

Proof. Let $X_\alpha \in \mathfrak{g}_\alpha - \{0\}$. The smallest Lie subalgebra containing $\mathfrak{g}_\alpha$ is

$$\langle \mathfrak{g}_\alpha \rangle := \bigcap_{\mathfrak{g}_\alpha \subset \mathfrak{h}} \{\mathfrak{h} \text{ is Lie subalgebra of } \mathfrak{g}\} = \mathfrak{g}_\alpha \oplus \mathbb{R} \cdot [X_\alpha, \overline{X_\alpha}];$$

cf. Lemma 10. Therefore $\langle \mathfrak{g}_\alpha \rangle$ is a three dimensional real vector space and it can easily be checked that $\langle \mathfrak{g}_\alpha \rangle$ and $\mathfrak{su}(2,\mathbb{C})$ are isomorphic as Lie algebras. Therefore, for any element $X \in \langle \mathfrak{g}_\alpha \rangle$, the closure of the one parameter group $\exp(\mathbb{R} \cdot X)$ is isomorphic

to a torus in $SU(2,\mathbb{C})$. Any torus in $SU(2,\mathbb{C})$ is isomorphic to the circle group S^1, and hence

$$\exp(\mathbb{R}\cdot X) \;\cong\; S^1.$$

The assertion for the orbits follows immediately from the identity

$$\mathrm{Ad}_{\exp X} = \exp(\mathrm{ad}_X) \quad \text{for all } X \in \mathfrak{g};$$

cf. [5, p. 23, Theorem 1.5.2]. □

5. More explicit description of the algorithm. To derive a more explicit description of the algorithm, it is necessary to take a closer look at the cost function (8a).

For $\Omega, X \in \mathfrak{g}$ and $t \in \mathbb{R}$, it is well known (cf. [5, p. 23]) that

$$\mathrm{Ad}_{\exp(t\Omega)}X = \exp(t\mathrm{ad}_\Omega)X = \sum_{k=0}^{\infty} \frac{1}{k!} t^k \mathrm{ad}_\Omega^k X. \tag{14}$$

The following convention is used to simplify notation. Let $\Omega \in \{E_\gamma, F_\gamma\}$ be one basis vector of the real root space $\mathfrak{g}_\gamma$ as in Proposition 7. Whenever "$\pm$" or "$\mp$" occurs in a formula, the upper sign stands for the case where $\Omega = E_\gamma$ while the lower one stands for the case where $\Omega = F_\gamma$. By Proposition 7 and Lemma 10 it holds that

$$\mathrm{ad}_{\mathfrak{g}_\gamma}(\mathfrak{t}) \subset \mathfrak{g}_\gamma \quad \text{and} \quad \mathrm{ad}_{\mathfrak{g}_\gamma}(\mathfrak{g}_\gamma) \subset \mathfrak{t}. \tag{15}$$

Therefore, by projecting (14) onto the torus algebra, one obtains (cf. (6))

$$\begin{aligned}
\mathrm{p}(\mathrm{Ad}_{\exp(t\Omega)}X) &= \sum_{k=0}^{\infty} \frac{1}{(2k)!} t^{2k}\ \mathrm{ad}_\Omega^{2k}\ X_0 + \sum_{k=0}^{\infty} \frac{1}{(2k+1)!} t^{2k+1}\ \mathrm{ad}_\Omega^{2k+1}\ c\cdot\overline{\Omega} \\
&= X_0 \mp \gamma(X_0)\cdot \sum_{k=0}^{\infty} \frac{1}{(2k+2)!} t^{2k+2}\ \mathrm{ad}_\Omega^{2k+1}\ \overline{\Omega} + \\
&\quad + c\cdot \sum_{k=0}^{\infty} \frac{1}{(2k+1)!} t^{2k+1}\ \mathrm{ad}_\Omega^{2k+1}\ \overline{\Omega},
\end{aligned}$$

where c denotes the $\overline{\Omega}$-coefficient of X. It is easily shown by induction that for all $k \in \mathbb{N}$

$$\mathrm{ad}_\Omega^{2k+1}\ \overline{\Omega} = \pm\Big(-\gamma(T_\gamma)\Big)^k\ T_\gamma \tag{16}$$

holds. A straightforward computation then leads to

$$\mathrm{p}(\mathrm{Ad}_{\exp(t\Omega)}X) = X_0 + g(t)\cdot T_\gamma, \tag{17}$$

where

$$g(t) := \frac{\gamma(X_0)}{\gamma(T_\gamma)} \cos\Big(\sqrt{\gamma(T_\gamma)}\cdot t\Big) - \frac{\gamma(X_0)}{\gamma(T_\gamma)} \pm \frac{c}{\sqrt{\gamma(T_\gamma)}} \sin\Big(\sqrt{\gamma(T_\gamma)}\cdot t\Big). \tag{18}$$

Because of Lemma 11, the cost function (8a), restricted to the orbit of a Lie algebra element $X \in \mathfrak{g}$ under the adjoint action of a one parameter group generated by some real root space element, is given by

$$f\Big|_{\mathrm{Ad}_{\exp(t\Omega)}X} = -\kappa(S,S) + \kappa\Big(X_0, X_0\Big) + \left(2g(t)\frac{\gamma(X_0)}{\gamma(T_\gamma)} + g(t)^2\right)\kappa\Big(T_\gamma, T_\gamma\Big). \tag{19}$$

From this expression we deduce an explicit formula for the step size selection (11). The following proposition and corollary are an adaptation of the results presented in section 8.4.1 of [14] to the Lie algebra setting.

PROPOSITION 19. *Let $X \in \mathfrak{g}$ and $\Omega \in \{\Omega_1, \ldots, \Omega_{2N}\}$ as in* (10) *be a basis vector of the root space $\mathfrak{g}_\gamma$. Let f denote the cost function* (8a). *Then, either*

$$f\Big|_{\mathrm{Ad}_{\exp(t\Omega)}X} \equiv f(X) \quad \textit{for all} \quad t \in \mathbb{R}$$

or

$$t \longmapsto f\Big|_{\mathrm{Ad}_{\exp(t\Omega)}X} \quad \textit{has periodicity} \quad \frac{\pi}{\sqrt{\gamma(T_\gamma)}}$$

and admits on

$$I := \left(-\frac{\pi}{2\sqrt{\gamma(T_\gamma)}}, \frac{\pi}{2\sqrt{\gamma(T_\gamma)}}\right]$$

exactly one minimum, namely at

$$t_*(X) = \begin{cases} \dfrac{\pi}{2\sqrt{\gamma(T_\gamma)}} & \textit{if} \quad \gamma(X_0) = 0, \\[2ex] \dfrac{1}{\sqrt{\gamma(T_\gamma)}} \arctan\left(\pm\dfrac{c\sqrt{\gamma(T_\gamma)}}{\gamma(X_0)}\right) & \textit{if} \quad \gamma(X_0) \neq 0, \end{cases} \tag{20}$$

where c denotes the $\overline{\Omega}$-coefficient of X.

Proof. The restricted cost function $f|_{\mathrm{Ad}_{\exp(t\Omega)}X}$ is constant if and only if $g(t)$ defined by (18) is constant, i.e.,

$$\begin{gathered} g'(t) \equiv 0 \\ \Longleftrightarrow \\ \pm\, c\,\sqrt{\gamma(T_\gamma)}\ \cos\left(\sqrt{\gamma(T_\gamma)}\cdot t\right) - \gamma(X_0)\ \sin\left(\sqrt{\gamma(T_\gamma)}\cdot t\right) \equiv 0 \\ \Longleftrightarrow \\ c = 0 \quad \text{and} \quad \gamma(X_0) = 0. \end{gathered}$$

Now let $c \neq 0$. From the identity

$$g(t) + g\left(t + \frac{\pi}{\sqrt{\gamma(T_\gamma)}}\right) = -2\frac{\gamma(X_0)}{\gamma(T_\gamma)}$$

one obtains after some computation

$$f\Big(\mathrm{Ad}_{\exp\left(\left(t+\frac{\pi}{\sqrt{\gamma(T_\gamma)}}\right)\Omega\right)}X\Big) - f\Big(\mathrm{Ad}_{\exp(t\Omega)}X\Big) = 0.$$

Computing the zeros $\widetilde{t} \in \mathbb{R}$ of $\frac{\mathrm{d}}{\mathrm{d}t} f(\mathrm{Ad}_{\exp(t\Omega)X})$ and the sign of the second derivative at $\widetilde{t}$ then completes the proof. ∎

Choosing the step size (20) in Ω-direction annihilates the $\overline{\Omega}$-component of X. More precisely we obtain the following.

COROLLARY 20. *Let $X \in \mathfrak{g}$ and $\Omega \in \mathfrak{B} = \{\Omega_1, \ldots, \Omega_{2N}\}$ be a basis vector of the root space $\mathfrak{g}_\gamma$, see* (10)*. Denote by*

$$\mathrm{p}_{\overline{\gamma}} : \mathfrak{g} \longrightarrow \mathbb{R} \cdot \overline{\Omega}$$

the projection onto the subspace $\mathbb{R} \cdot \overline{\Omega} \subset \mathfrak{g}_\gamma$. Then

$$\mathrm{p}_{\overline{\gamma}}(\mathrm{Ad}_{\exp(t\Omega)}X) = \left(\mp \frac{\gamma(X_0)}{\sqrt{\gamma(T_\gamma)}} \sin\left(\sqrt{\gamma(T_\gamma)}\, t\right) + c \cos\left(\sqrt{\gamma(T_\gamma)}\, t\right)\right) \overline{\Omega},$$

where c denotes the $\overline{\Omega}$-coefficient of X. Consequently, choosing the step size $t_*(X)$ as in (20) *annihilates the $\overline{\Omega}$-component of X.*

Proof. We use again (14) and (15) to deduce

$$\mathrm{p}_{\overline{\gamma}}(\mathrm{Ad}_{\exp(t\Omega)}X) = \sum_{k=0}^{\infty} \frac{1}{(2k+1)!} t^{2k+1}\, \mathrm{ad}_\Omega^{2k}\, \mathrm{ad}_\Omega X_0 + \sum_{k=0}^{\infty} \frac{1}{(2k)!} t^{2k}\, \mathrm{ad}_\Omega^{2k}\, c\, \overline{\Omega}.$$

It is easily seen by induction that $\mathrm{ad}_\Omega^{2k}\overline{\Omega} = (-\gamma(T_\gamma))^k \overline{\Omega}$ and it holds that $\mathrm{ad}_\Omega X_0 = \mp\gamma(X_0)\overline{\Omega}$; hence

$$\begin{aligned}
\mathrm{p}_{\overline{\gamma}}(\mathrm{Ad}_{\exp(t\Omega)}X) &= \mp\, \gamma(X_0) \sum_{k=0}^{\infty} \frac{\left(\sqrt{\gamma(T_\gamma)}\, t\right)^{2k+1}}{(2k+1)!} \frac{(-1)^k}{\sqrt{\gamma(T_\gamma)}}\, \overline{\Omega} \\
&\quad + c \sum_{k=0}^{\infty} \frac{t^{2k}}{(2k)!}\, (-\gamma(T_\gamma))^k \overline{\Omega} \\
&= \left(\mp \frac{\gamma(X_0)}{\sqrt{\gamma(T_\gamma)}} \sin\left(\sqrt{\gamma(T_\gamma)}\, t\right) + c \cos\left(\sqrt{\gamma(T_\gamma)}\, t\right)\right) \overline{\Omega}.
\end{aligned}$$

The last statement follows from a simple calculation by substituting $t_*(X)$ into the last equation. ∎

Note, that the Ω-component of X is not affected by the transformation $\mathrm{Ad}_{\exp(t\Omega)}X$, thus the two minimization steps along the Ω- and $\overline{\Omega}$-directions can be done simultaneously.

6. Convergence proof. We can now describe the main result of this paper. It is shown that the convergence of the Jacobi algorithm on compact Lie algebras is locally quadratically fast, provided the adjoint orbit $\mathcal{O}_S$ has maximal dimension. The dimension of $\mathcal{O}_S$ is equal to the dimension of the tangent space at $Z \in \mathcal{O}_S \cap \mathfrak{t}$. Now let $\alpha_1, \ldots, \alpha_k$ denote the roots for which

$$\alpha_i(Z) = 0, \qquad i = 1, \ldots, k,$$

holds. Hence $\mathrm{ad}_Z \mathfrak{g}_{\alpha_i} = 0$ for $i = 1, \ldots, k$ and therefore

$$\ker \mathrm{ad}_Z = \mathfrak{t} \oplus \sum_i \mathfrak{g}_{\alpha_i}$$

and

$$\dim \mathcal{O}_S = \dim T_Z\mathcal{O}_S = \dim\{\mathrm{ad}_Z H \mid H \in \mathfrak{g}\} = \dim \mathfrak{g} - \dim \mathfrak{t} - 2k.$$

This formula for the dimension justifies the following.

DEFINITION 21. *An element $S \in \mathfrak{g}$ is called* regular *if* $\dim \mathcal{O}_S = \dim \mathfrak{g} - \dim \mathfrak{t}$.

Example. The set of skew-Hermitian $(n \times n)$-matrices forms the Lie algebra $\mathfrak{u}(n, \mathbb{C})$. Fix a maximal Abelian subalgebra

$$\mathfrak{t} = \{T \in \mathfrak{u}(n, \mathbb{C}) \mid T = \mathrm{diag}(\mathrm{i}t_1, \ldots, \mathrm{i}t_n), t_j \in \mathbb{R}\}.$$

The roots in this case turn out to be $\alpha(T) = \pm(t_i - t_j)$ for $i < j$. So the regular elements of $\mathfrak{u}(n, \mathbb{C})$ are exactly those matrices having pairwise distinct eigenvalues.

The set of regular elements in $\mathfrak{g}$ is connected, open, and dense (cf. [5, p. 118 Theorem 2.8.5 and p. 146]). Therefore, the assumption in the following proposition is generically satisfied.

THEOREM 22 (main theorem). *An element $Z \in \mathcal{O}_S$ is a fixed point of the algorithm if and only if* $\mathrm{p}(Z) = Z$ *holds. If $S \in \mathfrak{g}$ is a regular element, the convergence of the Jacobi algorithm* (13) *is locally quadratically fast.*

Proof. The first statement of the theorem is implied by the following argument. Obviously, the only candidates for fixed points are critical points of the cost function. On the other hand, by Lemma 15, the algorithm is stationary neither at saddle points nor at global maxima as in both cases there exists at least one Ω_i-direction leading downhill.

For the proof of the convergence property, we will show that a sweep is smooth in a neighborhood of a minimum $Z \in \mathcal{O}_S$ of the cost function (8a). Furthermore, its derivative vanishes at Z and hence a simple Taylor argument will finish the proof of the local quadratic convergence.

In a first step, the smoothness of the step size selections (11) is shown. Let N denote the number of real root spaces and let

$$I := \left(-\frac{\pi}{2\sqrt{\gamma(T_\gamma)}}, \frac{\pi}{2\sqrt{\gamma(T_\gamma)}} \right].$$

For $i = 1, \ldots, 2N$ define

$$\phi_i : I \times \mathcal{O}_S \longrightarrow [0, \infty), \qquad \phi_i(t, X) = f(\mathrm{Ad}_{\exp(t\Omega_i)} X),$$
$$\psi_i : I \times \mathcal{O}_S \longrightarrow \mathbb{R}, \qquad \psi_i(t, X) = \mathrm{D}_1\phi_i(t, X),$$

where D_k denotes the first derivative with respect to the kth argument. By definition of $t_*^{(i)}$ (see (11)), it holds that

$$\psi_i\Big(t_*^{(i)}(X), X\Big) \equiv 0.$$

As in the proof of Lemma 14, one obtains for $\Omega_i \in \mathfrak{g}_\gamma$

$$\mathrm{D}_1\psi_i(t, X)\Big|_{(0,Z)} = -2\gamma(Z)^2\kappa(\Omega_i, \Omega_i) > 0.$$

This holds for all $\gamma \in \Sigma^+$ as S is a regular element. By continuity, $\mathrm{D}_1\psi_i(t, X)$ is greater than zero in a neighborhood of $(0, Z) \in I \times \mathcal{O}_S$. Hence the critical value

$$\phi_i(t_*^{(i)}(X), X)$$

is minimal for $X \in U$. This minimum is unique due to Proposition 19. Thus the implicit function theorem implies that the functions $t_*^{(i)}$ (11) are smooth in a neighborhood U of Z, $i = 1, \ldots, 2N$.

Let $\xi \in T_Z\mathcal{O}_S = \operatorname{span}(\Omega_1, \ldots, \Omega_{2N})$, the tangent space of $\mathcal{O}_S$ at Z. Then

$$\begin{aligned} \mathrm{D}_2\psi_i(t,X)\Big|_{(0,Z)}\xi &= -2\mathrm{D}_2\kappa(\Omega_i, \operatorname{ad}_{Z_0} Z)\xi \\ &= -2\Big(\kappa(\operatorname{ad}_{\Omega_i}\xi_0, Z) + \kappa(\operatorname{ad}_{\Omega_i} Z_0, \xi)\Big) \\ &= -2\kappa(\operatorname{ad}_{\Omega_i} Z, \xi) \end{aligned} \tag{21}$$

as $\xi_0 = \mathrm{p}(\xi) = 0$ and $Z_0 = \mathrm{p}(Z) = Z$. Any partial optimization step within a sweep is described by the mapping

$$r_i : \mathcal{O}_S \longrightarrow \mathcal{O}_S, \quad X \longmapsto \operatorname{Ad}_{\exp\left(t_*^{(i)}(X)\Omega_i\right)} X.$$

The derivative of r_i at Z acting on ξ is

$$\begin{aligned} \mathrm{D}r_i(Z)\xi &= \operatorname{Ad}_{\exp(t_*^{(i)}(Z)\Omega_i)}\xi + \Big(\operatorname{ad}_{\Omega_i}\operatorname{Ad}_{\exp(t_*^{(i)}(Z)\Omega_i)}(Z)\Big) \circ \mathrm{D}t_*^{(i)}(Z)\xi \\ &= \xi + \operatorname{ad}_{\Omega_i}(Z) \circ \mathrm{D}t_*^{(i)}(Z)\xi. \end{aligned}$$

By differentiating the equation

$$\psi_i\Big(t_*^{(i)}(X), X\Big) \equiv 0$$

with respect to X in direction ξ, one obtains by the chain rule

$$\mathrm{D}\psi_i\Big(t_*^{(i)}(Z), Z\Big)\xi = \mathrm{D}_1\psi_i\Big(t_*^{(i)}(Z), Z\Big) \cdot \mathrm{D}t_*^{(i)}(Z)\xi + \mathrm{D}_2\psi_i\Big(t_*^{(i)}(Z), Z\Big)\xi = 0.$$

Hence

$$\mathrm{D}t_*^{(i)}(Z)\xi = -\frac{\mathrm{D}_2\psi_i\Big(t_*^{(i)}(Z), Z\Big)}{\mathrm{D}_1\psi_i\Big(t_*^{(i)}(Z), Z\Big)}\xi = -\frac{\kappa(\operatorname{ad}_{\Omega_i} Z, \xi)}{\gamma(Z)^2\kappa(\Omega_i, \Omega_i)}.$$

The derivative for one partial step of the Jacobi sweep at Z therefore is

$$\begin{aligned} \mathrm{D}r_i(Z)\xi &= \xi - \operatorname{ad}_{\Omega_i} Z \frac{\kappa\Big(\operatorname{ad}_{\Omega_i} Z, \xi\Big)}{\gamma(Z)^2\kappa\Big(\Omega_i, \Omega_i\Big)} \\ &= \xi - \gamma(Z)\overline{\Omega}_i \frac{\kappa\Big(\gamma(Z)\overline{\Omega}_i, \xi\Big)}{\gamma(Z)^2\kappa\Big(\Omega_i, \Omega_i\Big)} \\ &= \xi - \overline{\Omega}_i \frac{\kappa\Big(\overline{\Omega}_i, \xi\Big)}{\kappa\Big(\Omega_i, \Omega_i\Big)}, \end{aligned}$$

where $\Omega_i \in \mathfrak{g}_\gamma$. It is easily seen that $\mathrm{D}r_i(Z)$ is a projection that annihilates the $\overline{\Omega}_i$-component of $\xi \in T_Z\mathcal{O}_S$. By the chain rule and the fact that Z is a fixed point of each partial step, i.e., $r_i(Z) = Z$ for all i, one calculates the derivative of one entire sweep operation

$$s(X) := \big(r_{2n} \circ r_{2n-1} \circ \cdots \circ r_2 \circ r_1\big)(X),$$

evaluated at the fixed point Z as

$$\mathrm{D}s(Z)\xi = \big(\mathrm{D}r_{2n} \circ \cdots \circ \mathrm{D}r_1\big)(Z)\xi = 0; \quad \text{therefore} \quad \mathrm{D}s(Z) \equiv 0. \tag{22}$$

Now choose open, relatively compact neighborhoods U, V of Z in $\mathcal{O}_S$, such that $s(U) \subset V$. U, V are diffeomorphic to open subsets of $\mathbb{R}^{2N}$ where $2N = \dim \mathcal{O}_S$. Without loss of generality, we may assume that U, V are open, bounded subsets of $\mathbb{R}^{2N}$. Reformulating everything in local coordinates, from Taylor's theorem, using $\mathrm{D}s(Z) \equiv 0$, we obtain

$$||s(X_k) - Z|| \leq \sup_{X \in \overline{U}} ||\mathrm{D}^2 s(X)|| \cdot ||X_k - Z||^2.$$

Thus the sequence $(X_k)_{k\in\mathbb{N}}$ generated by the Jacobi algorithm converges quadratically fast to Z. □

Theorem 22 generalizes the local convergence results for the Hermitian eigenvalue problem [21]. Our proof applies to any cyclic method and is not restricted to what is called a rowwise or columnwise cyclic method. The achieved shape of the "diagonalized" matrix need not necessarily be diagonal, but can be specified by the choice of the torus algebra. Furthermore, the theory is independent of choices of matrix representations of the underlying Lie algebra, hence a variety of structured matrix problems fit well into this setting. Several matrix representations of the classical Lie algebras can be found, e.g., in [12].

7. Pseudo code for the algorithm. Here we present a Matlab-like pseudo code for the algorithm. Our formulation is sufficiently general such that one can easily adapt the algorithm to any compact Lie algebra. Note that in section 8, the algorithm is examplified, using the Lie algebra $\mathfrak{sp}(n)$.

Let $\mathfrak{g}$ be a compact Lie algebra, $\mathfrak{t} \subset \mathfrak{g}$ a maximal Abelian subalgebra. Let a Lie algebra element $X \in \mathfrak{g}$ as well as a real root α be given. Denote by $\mathrm{p} : \mathfrak{g} \to \mathfrak{t}$ the orthogonal projection onto the torus algebra $\mathfrak{t}$. Let a basis of the corresponding root space $\mathfrak{g}_\alpha$ be $\{E_\alpha, F_\alpha\}$. Let this basis be normalized such that

$$\alpha(T_\alpha) = 1, \quad \text{where} \quad T_\alpha = [E_\alpha, F_\alpha]. \tag{23}$$

Then for $\Omega \in \{E_\alpha, F_\alpha\}$ the algorithm computes a pair $(\sin t, \cos t)$, such that $\exp(t\Omega)X\exp(-t\Omega)$ has no $\overline{\Omega}$-component. For the occurring $\pm$ signs see section 5. Using standard trigonometric formulas, one obtains for the step size selections $t_*^{(i)}(X)$ (cf. Proposition 19) the identities

$$\begin{aligned} \sin t_*(X) &= \pm\,\mathrm{sign}(\alpha(X_0)) \cdot \frac{c}{\sqrt{\alpha(X_0)^2 + c^2}}, \\ \cos t_*(X) &= \frac{|\alpha(X_0)|}{\sqrt{\alpha(X_0)^2 + c^2}}, \end{aligned} \tag{24}$$

where c denotes the $\overline{\Omega}$-coefficient of X and $X_0 = \mathrm{p}(X)$ is the orthogonal projection of X into the torus algebra.

Algorithm 1. Partial Step of Jacobi Sweep.

function: (cos, sin) = elementary.rotation(X, Ω)

Set $c :=$ $\overline{\Omega}$-component of X.

Set $S_0 := \mathrm{p}(X)$.
if $\alpha(X_0) \neq 0$
Set $(cos, sin) := \left(\frac{|\alpha(X_0)|}{\sqrt{\alpha(X_0)^2 + c^2}},\ \pm \mathrm{sign}(\alpha(X_0)) \cdot \frac{c}{\sqrt{\alpha(X_0)^2 + c^2}} \right)$.
else
if $c \neq 0$
Set $(cos, sin) := (0, 1)$.
else
Set $(cos, sin) := (1, 0)$.
endif
endif
end elementary.rotation

Denote by N the number of real roots and let

$$\mathfrak{B} = \{\Omega_1, \Omega_2, \ldots, \Omega_{2N}\}$$

be a basis of $\mathfrak{g}/\mathfrak{t}$ as in (10) normalized as in (23). Denote the real root corresponding to the basis $\Omega_{2i-1}, \Omega_{2i}$ by α_i and let f denote the cost function (8a). Given a Lie algebra element $S \in \mathfrak{g}$ and a tolerance $tol > 0$, this algorithm overwrites S by gSg^{-1}, where $g \in \exp(\mathfrak{g})$ and $f(gSg^{-1}) \leq tol$.

Algorithm 2. Jacobi Algorithm.
Set $g :=$ identity matrix.
while $f(S) > tol$
for $i = 1 : N$
$(cos, sin) :=$ elementary.rotation(S, Ω_{2i-1}).
$r_1 := \exp(t_*(S)\Omega_{2i-1})$.
$S := r_1 S r_1^{-1}$.
$(cos, sin) :=$ elementary.rotation(S, Ω_{2i}).
$r_2 := \exp(t_*(S)\Omega_{2i})$.
$S := r_2 S r_2^{-1}$.
$g := r_2^{-1} r_1^{-1} g$.
endfor
endwhile

8. Numerical experiments. We illustrate the approach by considering the task of finding the eigenvalues of a skew-Hermitian Hamiltonian matrix. As mentioned before, the set of skew-Hermitian, Hamiltonian matrices forms the compact Lie algebra $\mathfrak{sp}(n)$; see Example 5. This Lie algebra can be identified with the Lie algebra $\mathfrak{u}(n, \mathbb{H})$ of unitary quaternionic $(n \times n)$-matrices. Although our previous theory is coordinate free and independent of the choice of matrix representations, choosing explicit descriptions for the Lie algebra elements, leads to different forms of the numerical algorithm. To illustrate this phenomenon, consider $\mathfrak{sp}(n)$. The Lie algebra $\mathfrak{sp}(n)$ has different isomorphic matrix representations, such as, e.g.,

$$\mathfrak{sp}(n) = \left\{ \begin{bmatrix} A & B \\ -\overline{B} & \overline{A} \end{bmatrix} \in \mathbb{C}^{2n \times 2n} \mid A^* = -A, B^\top = B \right\},$$

or as in the example below. There are various natural choices for a torus algebra of $\mathfrak{sp}(n)$, e.g.,

$$\mathfrak{t} = \left\{ \begin{bmatrix} \mathrm{i}\Lambda & 0 \\ 0 & -\mathrm{i}\Lambda \end{bmatrix} \mid \Lambda \in \mathbb{R}^{n\times n} \text{ is diagonal} \right\},$$

$$\mathfrak{t}' = \left\{ \begin{bmatrix} 0 & \Lambda \\ -\Lambda & 0 \end{bmatrix} \mid \Lambda \in \mathbb{R}^{n\times n} \text{ is diagonal} \right\},$$

$$\mathfrak{t}'' = \left\{ \begin{bmatrix} 0 & \mathrm{i}\Lambda \\ \mathrm{i}\Lambda & 0 \end{bmatrix} \mid \Lambda \in \mathbb{R}^{n\times n} \text{ is diagonal} \right\},$$

leading to isomorphic variants of the eigenvalue problem. More matrix representations of classical Lie algebras can be found in [12].

As a computational example, we consider the eigenvalue problem for an isomorphic copy of $\mathfrak{sp}(n)$. Thus let

$$\mathfrak{g} = \left\{ \begin{bmatrix} A & B & C & D \\ -B & A & D & -C \\ -C & -D & A & B \\ -D & C & -B & A \end{bmatrix} ; A, B, C, D \in \mathbb{R}^{n\times n} \right| \tag{25}$$
$$\left. A^\top = -A, B^\top = B, C^\top = C, D^\top = D \right\}.$$

Note that $\mathfrak{g}$ is isomorphic to $\mathfrak{sp}(n)$ via the real Lie algebra isomorphism

$$\rho : \mathfrak{sp}(n) \longrightarrow \mathfrak{g}, \qquad X \longmapsto \begin{bmatrix} \mathrm{Re}X & \mathrm{Im}X \\ -\mathrm{Im}X & \mathrm{Re}X \end{bmatrix}. \tag{26}$$

Let $\otimes$ denote the Kronecker product. The torus algebra of $\mathfrak{g}$ is chosen as

$$\mathfrak{t} = \left\{ \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \otimes C_{\mathrm{diag}} \mid C_{\mathrm{diag}} \in \mathbb{R}^{n\times n} \text{ is diagonal} \right\}. \tag{27}$$

If $\pm\mathrm{i}\lambda_k$ are the eigenvalues of the skew-Hermitian Hamiltonian matrix, the entries of the diagonal matrix C_{diag} in (27) consist of the λ_k's. Let $C_{\mathrm{diag}} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$. With the assumptions above one computes the n^2 real roots as

$$\boxed{\begin{array}{ll} \lambda_i - \lambda_j, & 1 \le i < j \le n, \\ \\ \lambda_i + \lambda_j, & 1 \le i \le j \le n. \end{array}}.$$

Hence the matrices are regular in the sense of Definition 21 if and only if the moduli of the λ_k's are pairwise distinct and $\lambda_k \neq 0$ for all k.

Let $E_{ij} \in \mathbb{R}^{n\times n}$ have (i,j)-entry equal to 1 and 0 elsewhere. As an orthogonal basis for the corresponding real root spaces that satisfies condition (23), choose

$$\mathfrak{B}_{\lambda_i-\lambda_j} = \left\{ \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \otimes (E_{ij}-E_{ji}),\ \frac{1}{2}\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \otimes (E_{ij}+E_{ji}) \right\},$$

$$\mathfrak{B}_{\lambda_i+\lambda_j} = \left\{ \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \otimes (E_{ij}+E_{ji}),\ \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \otimes (E_{ij}+E_{ji}) \right\}.$$

Then we obtain an ordered basis $\mathfrak{B}$ of $\mathfrak{sp}(n)/\mathfrak{t}$ as in (10) via

$$\mathfrak{B} = \bigcup_{1\le i<j\le n} \mathfrak{B}_{\lambda_i-\lambda_j} \cup \bigcup_{1\le i\le j\le n} \mathfrak{B}_{\lambda_i+\lambda_j}. \tag{28}$$

Now let $X \in \mathfrak{g}$ in the chosen representation (25). Let Ω_k be the kth element in $\mathfrak{B}$; cf. (28). Denote by $X_{[r,s]}$ the (r,s)-entry of X. Then the algorithms of section 7 are explicitly as follows. Algorithm 1′ computes the nontrivial entries $(\sin t_*(X), \cos t_*(X)$ occurring in the matrix $\exp(t_*(X)\Omega_k)$; cf. (11) and (24). Hence in Algorithm 2′, the matrix $\exp(t_*(S)\Omega_k)$ need not be calculated explicitly but can easily be constructed by replacing the required entries with the computed $\sin t_*(X)$ and $\cos t_*(X)$), respectively.

ALGORITHM 1′. PARTIAL STEP OF JACOBI SWEEP.

function: (cos, sin) = elementary.rotation(X, Ω_k)
  if $1 \le k \le n^2 - n$
    Set $\alpha(X_0) := X_{[i,2n+i]} - X_{[j,2n+j]}$.
    if k is odd
      Set $c := 2X_{[i,2n+j]}$.
    else
      Set $c := -2X_{[i,j]}$.
    endif
  else
    Set $\alpha(X_0) := X_{[i,2n+i]} + X_{[j,2n+j]}$.
    if k is odd
      Set $c := 2X_{[i,3n+j]}$.
    else
      Set $c := -2X_{[i,n+j]}$.
    endif
  endif
  if $\alpha(X_0) \neq 0$
    Set $(cos, sin) := \left(\dfrac{|\alpha(X_0)|}{\sqrt{\alpha(X_0)^2 + c^2}},\ \operatorname{sign}(\alpha(X_0)) \cdot \dfrac{c}{\sqrt{\alpha(X_0)^2 + c^2}} \right)$.
  else
    if $c \neq 0$
      Set $(cos, sin) := (0, 1)$.
    else
      Set $(cos, sin) := (1, 0)$.
    endif
  endif
end elementary.rotation

Given a Lie algebra element $S \in \mathfrak{g}$ and a tolerance $tol > 0$, the following algorithm overwrites S by $gSg^\top$, where $g \in \exp(\mathfrak{g})$. Since the Killing form on $\mathfrak{g}$ is given by

$$\kappa(X, Y) = 4(n+1)\mathrm{tr}(XY),$$

the cost function (8a) is

$$f(S) = -4(n+1)\mathrm{tr}\Big((S - S_0)^2\Big),$$

where S_0 denotes the projection of S onto $\mathfrak{t}$.

ALGORITHM 2′. JACOBI ALGORITHM.
Set $g :=$ identity matrix.
while $f(S) > tol$
  for $k = 1 : 2n^2$
    $(cos, sin) :=$ elementary.rotation(S, Ω_k).
    Set $r := \exp(t_*(S)\Omega_k)$.
    Set $g := r^\top g$.
  endfor
endwhile

Finally, some numerical experiments are presented which are compatible with local quadratic convergence. All simulations are done using MATHEMATICA 4.0; cf. [39]. For a given torus algebra element T, the initial point S is generated in the following way. Let $\Omega_k \in \mathfrak{B}$ (cf. (10)), an ordered basis of $\mathfrak{g}$, where $n := 15$. Then $\dim \mathfrak{g} = 465$ real values $t_1, \ldots, t_{465} \in [-\pi, \pi]$ are chosen by using the MATHEMATICA-command Random. A generic group element g is generated via

$$g = \prod_{k=1}^{465} \exp(t_k \Omega_k).$$

The initial point S is obtained by conjugating T with g, namely $S = gTg^\top$. Every experiment is done with three different randomly chosen initial points, plotted together in one diagram where the value of the cost function is on the vertical axes. The following simulations have been done.

Figure 8.1	$C_{\mathrm{diag}} = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)$
Figure 8.2	$C_{\mathrm{diag}} = (96, 97, 97.5, 98, 98.5, 99, 99.5, 100, 100.5, 101, 101.5, 102, 102.5, 103, 104)$
Figure 8.3	$C_{\mathrm{diag}} = (0, 0, 0, 10, 10, 10, 20, 20, 20, 30, 30, 30, 40, 40, 50)$
Figure 8.4	$C_{\mathrm{diag}} = (99.9998, 100.001, 100.0002, 100.03, 100.002, 100.001, 99.997, -0.002, 0.01, 0.2, -0.03, -0.001, 0.01, 0.002, 0.0001)$

For the simulation in Figure 8.2, the absolute values of all eigenvalues are close to 100. Nonregular elements show the same convergence behavior; cf. Figure 8.3. Figure 8.4 illustrates the convergence behavior of the algorithm in the case when there is a gap between the eigenvalues.

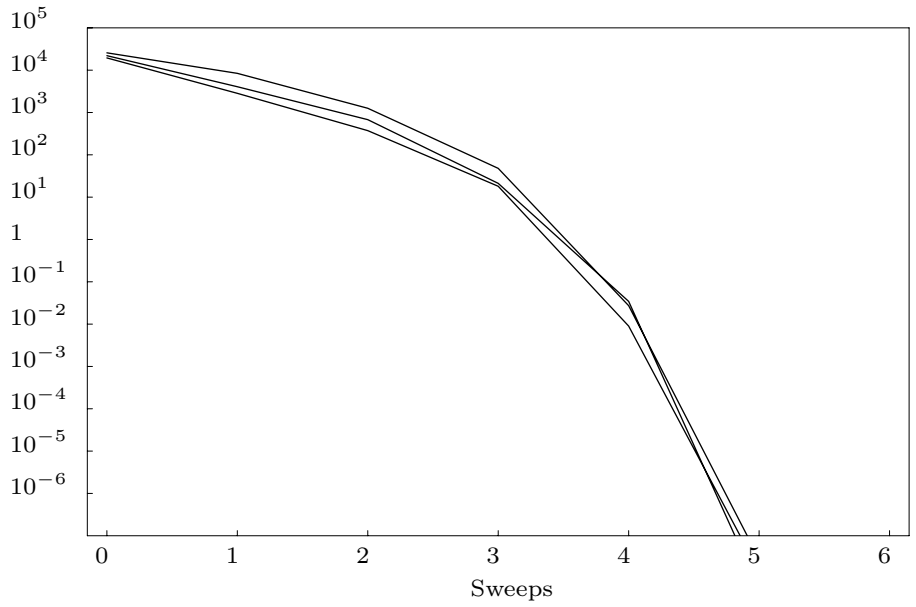


FIG. 8.1. *Convergence behavior for a regular element.* $\dim \mathfrak{g} = 465$, $f = -\kappa(X - X_0, X - X_0)$.

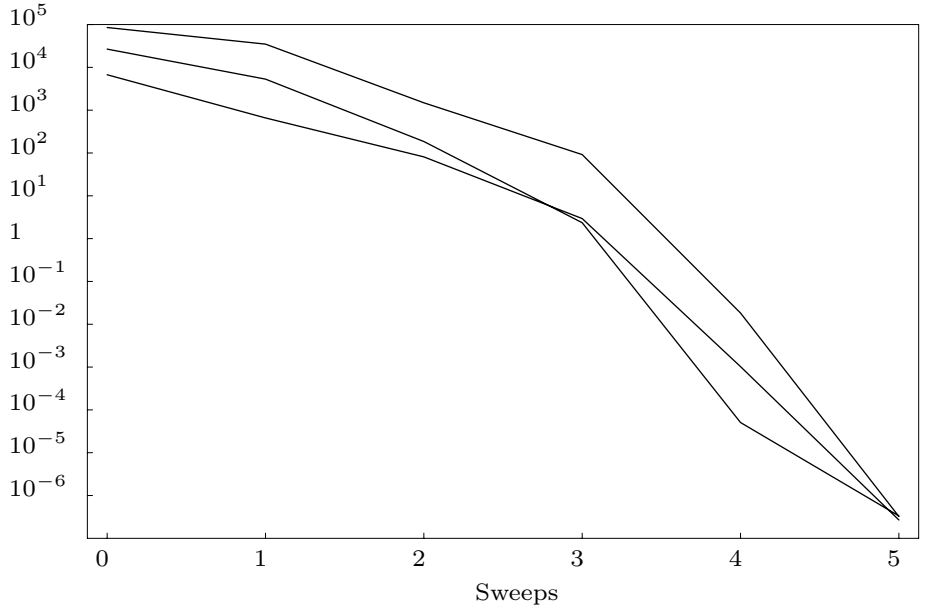


FIG. 8.2. *Convergence behavior for an element with eigenvalues near* 100, -100, *respectively,* $\dim \mathfrak{g} = 465$, $f = -\kappa(X - X_0, X - X_0)$.

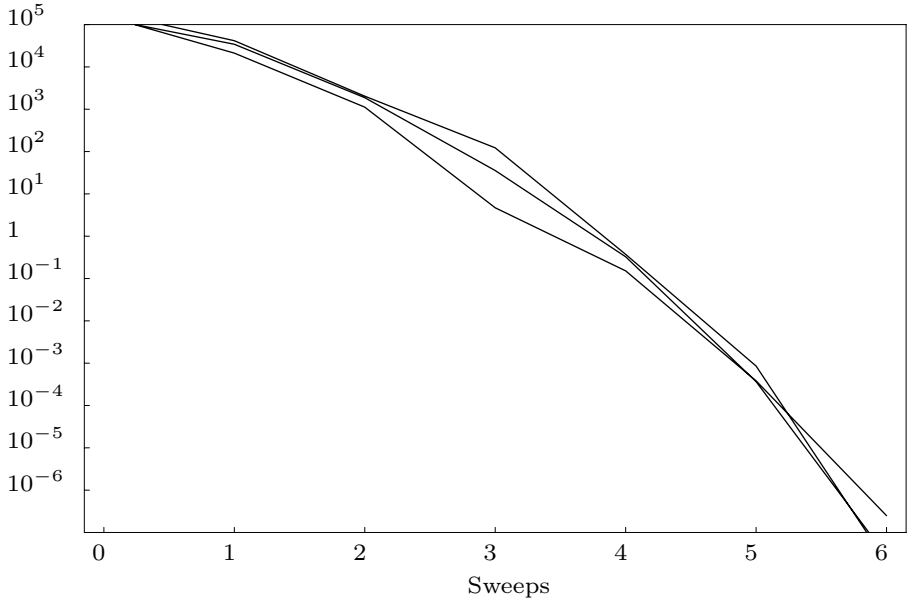


FIG. 8.3. *Convergence behavior for a nonregular element.* $\dim \mathfrak{g} = 465$, $f = -\kappa(X - X_0, X - X_0)$.

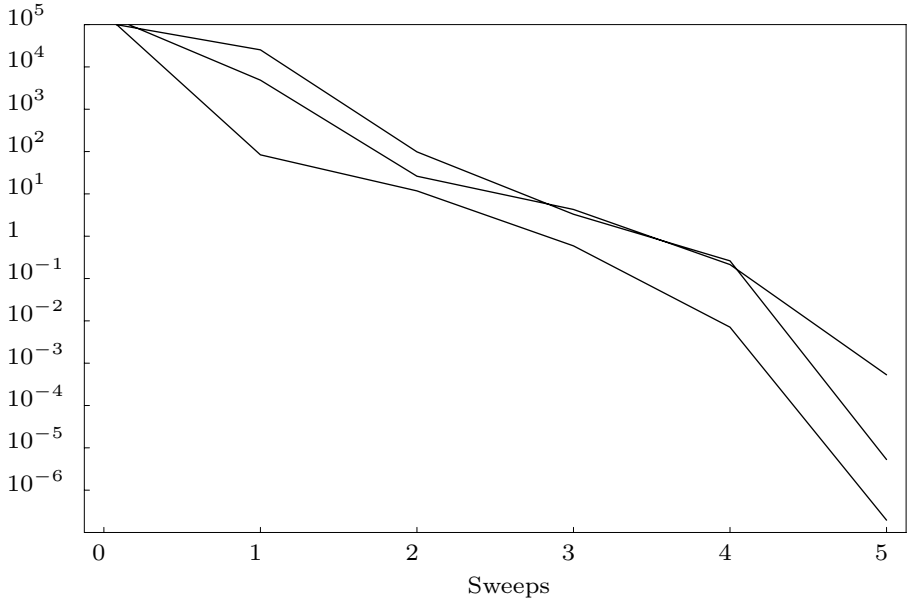


FIG. 8.4. *Convergence behavior for an element with a gap between great and small eigenvalues.* $\dim \mathfrak{g} = 465$, $f = -\kappa(X - X_0, X - X_0)$.

Acknowledgment. The first author thanks G. Dirr and L. Kramer, Würzburg, for some fruitful discussions.

REFERENCES

[1] T. Bröcker and T. tom Dieck, *Representations of Compact Lie Groups*, Grad. Texts in Math 98, Springer-Verlag, New York, 1995.
[2] A. Bunse-Gerstner and H. Fassbender, *On the generalized Schur Decomposition of a matrix pencil for parallel computation*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 911–939.
[3] R. Byers, *A Hamiltonian-Jacobi algorithm*, IEEE Trans. Automat. Control, 35 (1990), pp. 566–570.
[4] J.-P. Charlier and P. Van Dooren, *A Jacobi-like algorithm for computing the generalized Schur form of a regular pencil*, J. Comput. Appl. Math., 27 (1989), pp. 17–36.
[5] J. J. Duistermaat and J. A. C. Kolk, *Lie Groups*, Springer-Verlag, Berlin, 2000.
[6] P. J. Eberlein, *A Jacobi-like method for the automatic computation of eigenvalues and eigenvectors of an arbitrary matrix*, J. Soc. Indust. Appl. Math., 10 (1962), pp. 74–88.
[7] P. J. Eberlein, *Solution to the complex eigenproblem by a norm reducing Jacobi type method*, Numer. Math., 14 (1990), pp. 232–245.
[8] P. J. Eberlein, *On the diagonalization of complex symmetric matrices*, J. Inst. Math. Appl., 7 (1971), pp. 377–383.
[9] H. Fassbender, D. S. Mackey, and N. Mackey, *Hamilton and Jacobi come full circle: Jacobi algorithms for structured Hamiltonian eigenproblems*, Linear Algebra Appl., 332/334 (2001), pp. 37–80.
[10] J. Fogarty, *Invariant Theory*, W. A. Benjamin Inc., New York, Amsterdam 1969.
[11] 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.
[12] R. Gilmore, *Lie Groups, Lie Algebras, and Some of Their Applications*, Wiley, New York, 1974.
[13] H. H. Goldstine and L. P. Horwitz, *A procedure for the diagonalization of normal matrices*, J. ACM, 6 (1959), pp. 176–195.
[14] G. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.
[15] G. Gose, *Das Jacobi-Verfahren für* $Ax = \lambda Bx$, ZAMM, 59 (1979), pp. 93–101.
[16] D. Hacon, *Jacobi's method for skew-symmetric matrices*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 619–628.
[17] W. Hein, *Einfuhrung in Struktur- und Darstellungstheorie der klassischen Gruppen*, Springer-Verlag, Berlin, 1990.
[18] U. Helmke and K. Hüper, *The Jacobi Method: A Tool for Computation and Control*, in Systems and Control in the Twenty-First Century, C. I. Byrnes, B. N. Datta, D. S. Gilliam, and C. F. Martin, eds., Birkhäuser Boston, Boston, 1997, pp. 205–228.
[19] U. Helmke and K. Hüper, *A Jacobi-type method for computing balanced realizations*, Systems Control Lett., 39 (2000), pp. 19–30.
[20] U. Helmke, K. Hüper, and J. B. Moore, *Computation of Signature Symmetric Balanced Realizations*, J. Global Optim., 27 (2003), pp. 135–148.
[21] P. Henrici, *On the speed of convergence of cyclic and quasicyclic Jacobi methods for computing eigenvalues of Hermitian matrices*, J. Soc. Indust. Appl. Math., 6 (1958), pp. 144–162.
[22] K. Hüper, *A Calculus Approach to Matrix Eigenvalue Algorithms*, Habilitation thesis, Department of Mathematics, Würzburg University, Germany, 2002.
[23] K. Hüper, *A Dynamical System Approach to Matrix Eigenvalue Algorithms*, in Mathematical Systems Theory in Biology, Communications, Computation, and Finance, J. Rosenthal and D. S. Gilliam, eds., IMA Vol. Math. Appl. 134, Springer-Verlag, 2003, pp. 257–274.
[24] K. Hüper, *Structure and Convergence of Jacobi-Type Methods for Matrix Computations*, Ph.D. thesis, Technical University of Munich, 1996.
[25] C. G. J. Jacobi, *Über ein leichtes Verfahren, die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen*, Crelle's J. für die reine und angewandte Mathematik, 30 (1846), pp. 51–94.
[26] A. W. Knapp, *Lie Groups Beyond an Introduction*, Birkhäuser Boston, Boston, 1996.
[27] E. G. Kogbetliantz, *Solution of linear equations by diagonalization of coefficient matrix*, Quart. Appl. Math., 13 (1955), pp. 123–132.
[28] N. Mackey, *Hamilton and Jacobi meet again: Quaternions and the eigenvalue problem*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 421–435.

[29] W. F. Mascarenhas, *A note on Jacobi being more accurate than QR*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 215–218.
[30] M. H. C. Paardekooper, *An eigenvalue algorithm for skew-symmetric matrices*, Numer. Math., 17 (1971), pp. 189–202.
[31] N. H. Rhee and V. Hari, *On the cubic convergence of the Paardekooper method*, BIT, 35 (1995), pp. 116–132.
[32] A. Schönhage, *Zur Konvergenz des Jacobi Verfahrens*, Numer. Math., 3 (1961), pp. 374–380.
[33] A. Schönhage, *Zur quadratischen Konvergenz des Jacobi-Verfahrens*, Numer. Math., 6 (1964), pp. 410–412.
[34] G. W. Stewart, *A Jacobi-like algorithm for computing the Schur decomposition of a nonhermitian matrix*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 853–864.
[35] H. P. M. van Kempen, *On the quadratic convergence of the special cyclic Jacobi method*, Numer. Math., 9 (1966), pp. 19–22.
[36] K. Veselic, *On a class of Jacobi-like procedures for diagonalizing arbitrary real matrices*, Numer. Math., 33 (1979), pp. 157–172.
[37] K. Veselic, *A Jacobi eigenreduction algorithm for definite matrix pairs*, Numer. Math., 64 (1993), pp. 241–269.
[38] N. J. Wildberger, *Diagonalization in compact Lie algebras and a new proof of a theorem of Kostant*, Proc. Amer. Math. Soc., 119 (1993), pp. 649–655.
[39] S. Wolfram, *The Mathematica Book. Version* 4.0, Cambridge University Press, Cambridge, UK, 1999.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 1, pp. 70–96

CUBICALLY CONVERGENT ITERATIONS FOR INVARIANT SUBSPACE COMPUTATION*

P.-A. ABSIL†, R. SEPULCHRE‡, P. VAN DOOREN§, AND R. MAHONY¶

Abstract. We propose a Newton-like iteration that evolves on the set of fixed dimensional subspaces of $\mathbb{R}^n$ and converges locally cubically to the invariant subspaces of a symmetric matrix. This iteration is compared in terms of numerical cost and global behavior with three other methods that display the same property of cubic convergence. Moreover, we consider heuristics that greatly improve the global behavior of the iterations.

Key words. invariant subspace, Grassmann manifold, cubic convergence, symmetric eigenproblem, inverse iteration, Rayleigh quotient, Newton method, global convergence

AMS subject classification. 65F15

DOI. 10.1137/S0895479803422002

1. Introduction.

The problem of computing a p-dimensional eigenspace (i.e., invariant subspace) of an $n \times n$ matrix $A = A^T$ is ubiquitous in applied mathematics, with applications in control theory, pattern recognition, data compression and coding, antenna array processing, and a multitude of other domains.

Several methods for subspace estimation were proposed in the late seventies and early eighties. Demmel [Dem87] provides a joint analysis of three of the early methods that refine initial estimates of arbitrary p-dimensional eigenspaces of a (possibly nonsymmetric) $n \times n$ data matrix A. The early methods depend on the various numerical solutions of a common Riccati equation. These methods converge at best quadratically (Chatelin's Newton-based method [Cha84]) even when A is symmetric and involve the solution of a Sylvester equation at each iteration step. Moreover, the iterations defined depend on a choice of normalization condition used to generate the Riccati equation as well as the present iterative estimate of the eigenspace. More recently, iterations have been proposed that operate "intrinsically" on the Grassmann manifold, the set of p-planes in $\mathbb{R}^n$. Watkins and Elsner [WE91] have studied a multishifted QR algorithm that, as we will show, conceals a Grassmannian generalization of the Rayleigh quotient iteration (RQI). Edelman, Arias, and Smith [EAS98] derived a Newton iteration directly on the Grassmann manifold to find critical points of a generalized Rayleigh quotient. A practical implementation of this method was investigated

*Received by the editors January 27, 2003; accepted for publication (in revised form) by L. Eldén October 30, 2003; published electronically August 6, 2004. An abridged version of this article appeared in the Proceedings of the 42nd IEEE Conference on Decision and Control.

http://www.siam.org/journals/simax/26-1/42200.html

†School of Computational Science and Information Technology, Florida State University, Tallahassee, FL 32306-4120 (absil@csit.fsu.edu). Part of this work was done while the author was a Research Fellow with the Belgian National Fund for Scientific Research (Aspirant du F.N.R.S.) at the University of Liège (www.montefiore.ulg.ac.be/∼absil). Part of this work was also done while the author was a guest at the Mathematisches Institut der Universität Würzburg under a grant from the European Nonlinear Control Network.

‡Department of Electrical Engineering and Computer Science, Université de Liège, Institut Montefiore (B28), Grande Traverse 10, B-4000 Liège, Belgium (r.sepulchre@ulg.ac.be, www.montefiore.ulg.ac.be/systems). The research of this author was partially supported by US Air Force grants F49620-01-1-0063 and F49620-01-1-0382.

§Department of Mathematical Engineering, Université Catholique de Louvain, Bâtiment Euler (A.119), Avenue Georges Lemaître 4, 1348 Louvain-la-Neuve, Belgium (vdooren@csam.ucl.ac.be).

¶Department of Engineering, Australian National University, Canberra, ACT, 0200, Australia.

by Lundström and Eldén [LE02]. In a recent paper [AMSV02], the authors proposed a generalization of the RQI posed directly on the Grassmann manifold where scalar shifts are replaced by a matrix shift. All these algorithms are intrinsically defined on the Grassmann manifold (i.e., the next iterate only depends on A and the current iterate) and converge locally cubically to the isolated p-dimensional eigenspaces of $A = A^T$.

In the present paper, we compare the three recently proposed cubically convergent iterations [WE91, EAS98, LE02, AMSV02] and propose a fourth cubically convergent method inspired by the multihomogeneous Newton methods considered by Dedieu and Shub [DS00]. The first goal of this paper is to compare the four iterations in terms of numerical cost and global behavior. The global behavior of these iterations is of particular interest as existing analytical results focus on the local convergence rates. In the case where $p = 1$ and only a single eigenvector is computed the three recently proposed methods degenerate to the same iteration, the classical RQI, for which the global behavior is well understood [PK69, Par80, BS89, PS95]. In contrast, almost no global analysis has been undertaken for the various iterations when $p > 1$. In this paper, we show that although the local performance of the methods is comparable, the global performance differs appreciably. In particular, we study for each method how the shape of the basin of attraction of an eigenspace deteriorates when some eigenvalues of A are clustered.

The second goal of this paper is to propose modifications to the methods that improve the global performance of the iterations without compromising the local performance. The purpose of the modifications is to ensure that each given eigenspace is surrounded by a large basin of attraction. This guarantees that the iteration converges to the targeted eigenspace even when started rather far away from it. For the Grassmannian RQI of [AMSV02] we propose a simple threshold on the distance between successive iterates that improves the shape of the basins of attraction. For the two Newton-based methods, we introduce a deformation parameter τ that achieves a continuous transition between the original iteration and a gradient flow with large basins of attraction. This deformation technique is related to line search methods and trust region methods in optimization. We propose a simple choice for τ that dramatically enlarges the basins of attraction around the attractors while preserving cubic convergence. In the case of the new Newton-like iteration proposed in this paper, the resulting algorithm (Algorithm 5.2) displays an excellent global behavior, combined with a cubic rate of convergence and a numerical cost of $O(np^2)$ flops per iteration when A is suitably condensed.

This paper is organized as follows. After a short review of subspaces, eigenspaces and their representations (section 2), we state four cubically convergent iterative algorithms for eigenspace computation (section 3). These iterations are compared in terms of numerical cost and global behavior in section 4. In section 5, we propose ways of improving the global behavior of the iterations. The main results are summarized in the concluding section 6.

2. Subspaces and eigenspaces. In the present section, we introduce concepts and notation pertaining to subspaces and eigenspaces.

Unless otherwise stated, all scalars, vectors, and matrices are real. The superscript ^{T} denotes the transpose. Following conventions in [HM94], we use $\mathrm{Grass}(p, n)$ to denote the *Grassmann manifold* of the p-dimensional subspaces of $\mathbb{R}^n$, $\mathbb{RP}^{n-1} = \mathrm{Grass}(1, n)$ to denote the real projective space, and $\mathrm{ST}(p, n)$ to denote the *noncompact Stiefel manifold*, i.e., the set of $n \times p$ matrices with full rank. The columns space of

$Y \in \mathrm{ST}(p,n)$ is denoted by $\mathrm{span}(Y)$. The "span" mapping is an application on $\mathrm{ST}(p,n)$ onto $\mathrm{Grass}(p,n)$ that is nowhere invertible. Given a matrix Y in $\mathrm{ST}(p,n)$, the set of matrix representations of the subspace $\mathrm{span}(Y)$ is

$$\mathrm{span}^{-1}(\mathrm{span}(Y)) = Y\,\mathrm{GL}_p := \{YM : M \in \mathrm{GL}_p\},$$

where GL_p denotes the set of $p \times p$ invertible matrices. This identifies $\mathrm{Grass}(p,n)$ with $\mathrm{ST}(p,n)/\mathrm{GL}_p := \{Y\,\mathrm{GL}_p : Y \in \mathrm{ST}(p,n)\}$. More details on the Grassmann manifold and matrix representations can be found in [FGP94, AMS02, Abs03].

Let A be an $n \times n$ matrix. Let $\mathcal{X}$ be a p-dimensional subspace of $\mathbb{R}^n$ and let $Q = [X|X_\perp]$ be an orthogonal $n \times n$ matrix such that X spans $\mathcal{X}$. Then Q^TAQ may be partitioned in the form $Q^TAQ = \left(\begin{smallmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{smallmatrix}\right)$ where $A_{11} \in \mathbb{R}^{p\times p}$. The subspace $\mathcal{X}$ is an *eigenspace* (i.e., invariant subspace) of A if and only if $A_{21} = 0$. By *spectrum* of $\mathcal{X}$, we mean the set of eigenvalues of A_{11}. The *external gap* of the eigenspace $\mathcal{X}$ of A is the shortest distance between the eigenvalues of A_{11} and the eigenvalues of A_{22}. The *internal gap* of $\mathcal{X}$ is the shortest distance between two eigenvalues of A_{11}. We say that $\mathcal{X}$ is a *nondefective* eigenspace of A if A_{11} is nondefective. The eigenspace $\mathcal{X}$ is termed *spectral* [RR02] if A_{11} and A_{22} have no eigenvalue in common (i.e., nonvanishing external gap). When $A = A^T$, an eigenspace is spectral if and only if it is *isolated*, i.e., there exists a ball in $\mathrm{Grass}(p,n)$ centered on $\mathcal{V}$ that does not contain any eigenspace of A other than $\mathcal{V}$. The span of a full-rank $n \times p$ matrix Y is an eigenspace of A if and only if there exists a matrix L such that $AY = YL$, in which case Y is called an *eigenbasis* and L the corresponding *eigenblock* [JS01].

From now on, we assume that $A = A^T$ unless otherwise specified.

3. Four iterations for eigenspace computation. In this section, we define four iterations that evolve on the Grassmann manifold of p-planes in $\mathbb{R}^n$ and converge locally cubically to the spectral eigenspaces of a symmetric $n \times n$ matrix A.

3.1. Shifted inverse iterations. Inverse iteration is a widely used method for computing eigenvectors of A corresponding to selected eigenvalues for which an approximation is available [Ips97]. Let $\hat{\lambda}$ be an approximation to an eigenvalue of A. Inverse iteration generates a sequence of vectors x_k starting from an initial vector x_0 by solving the systems of linear equations

$$(A - \hat{\lambda}I)z = x_k \tag{3.1}$$

and usually normalizing the result $x_{k+1} := z/\|z\|$. From a theoretical point of view, the norm of x_k is irrelevant: the iteration (3.1) induces an iteration on the projective space, i.e., the set of one-dimensional subspaces of $\mathbb{R}^n$. Except in some nongeneric cases, the iteration converges to an eigenvector of A with an eigenvalue closest to $\hat{\lambda}$, and the rate of convergence is linear. However, a higher rate of convergence can be achieved by adapting $\hat{\lambda}$ "online" using the information given by the current iterate x_k. For $A = A^T$, the choice of the feedback law $\hat{\lambda} := \rho(x_k)$, where ρ denotes the Rayleigh quotient

$$\rho(y) := \frac{y^TAy}{y^Ty}, \tag{3.2}$$

yields the well-known RQI

$$(A - \rho(x_k)I)z = x_k,\ x_{k+1} = z/\|z\|. \tag{3.3}$$

The fact that ρ provides a quadratic approximation of an eigenvalue around the corresponding eigenvector explains why the rate of convergence is lifted from linear to cubic [Par80, AMSV02].

In the present paper, we consider two ways of generalizing the RQI to the Grassmann manifold of p-planes in $\mathbb{R}^n$. The first possibility is to use multiple scalar shifts, where the shifts are the Ritz values computed from the current subspace.

ALGORITHM 3.1 (RSQR). *Iterate the mapping* $\mathrm{Grass}(p,n) \ni \mathcal{Y} \mapsto \mathcal{Y}_+ \in \mathrm{Grass}(p,n)$ *defined by*

1. *Pick an orthonormal* $n \times p$ *matrix* Y *that spans* $\mathcal{Y}$.
2. *Solve for* $Z \in \mathbb{R}^{n\times p}$ *the equation*

$$(A-\rho_1 I)\dots(A-\rho_p I)Z = Y, \tag{3.4}$$

where $\rho_1, \dots, \rho_p$ *are the eigenvalues of* Y^TAY *repeated according to their multiplicity.*

3. *Define* $\mathcal{Y}_+$ *as the span of* Z.

We call this iteration *RSQR* because of its link with the generalized Rayleigh-shifted QR algorithm studied in [WE91]. It comes as a corollary from the results of [WE91] that RSQR converges locally cubically to the spectral eigenspaces of $A = A^T$, as we now explain.

The RQI algorithm is related to the Rayleigh-shifted QR algorithm, as shown e.g., in the enlightening paper by Watkins [Wat82]. The QR algorithm on the matrix A with Rayleigh quotient shift can be written as a QR decomposition

$$(A-\sigma_k I)Q_k = Q_{k+1}R_{k+1}, \tag{3.5}$$

where σ_k is the lower right element of $A_k = Q_k^TAQ_k$. Taking the inverse transpose of (3.5) yields, assuming $A = A^T$,

$$(A-\sigma_k I)^{-1}Q_k = Q_{k+1}R_{k+1}^{-T}, \tag{3.6}$$

where R_{k+1}^{-T} is now a lower triangular matrix. The last column of (3.6) yields

$$(A-\sigma_k I)^{-1}x_k = r_{k+1}^{-1}x_{k+1},$$

where x_k denotes the last column of Q_k and r_k denotes the lower right element of R_k. This is RQI (3.3). In [WE91], Watkins and Elsner study a *generalized Rayleigh-quotient shift strategy* for the QR algorithm. It consists in replacing $(A-\sigma_k I)$ by $\mathcal{P}(A)$, where $\mathcal{P}(\lambda)$ is the characteristic polynomial of the $p \times p$ lower right submatrix of $Q_k^TAQ_k$. In this case, (3.5) becomes

$$\mathcal{P}(A)Q_k = Q_{k+1}R_{k+1} \tag{3.7}$$

or equivalently, taking the inverse transpose,

$$\mathcal{P}(A)^{-1}Q_k = Q_{k+1}R_{k+1}^{-T}$$

whose last p columns yield

$$\mathcal{P}(A)^{-1}X_k = X_{k+1}L_{k+1}.$$

Here X_k denotes the last p columns of Q_k and $\mathcal{P}(A) := (A-\rho_1 I)\dots(A-\rho_p I)$, where $\rho_1, \dots, \rho_p$ denote the eigenvalues of $X_k^TAX_k$. This iteration maps the span of

X_k to the span of X_{k+1}, and this is the above-defined RSQR (Algorithm 3.1). The developments in [WE91] show that this iteration converges locally cubically to the spectral eigenspaces of A. That is, for each spectral eigenspace $\mathcal{V}$ of A, there exist a scalar c and a neighborhood B such that $\mathrm{dist}(\mathcal{Y}_+, \mathcal{V}) \leq c\,\mathrm{dist}(\mathcal{Y}, \mathcal{V})^3$ for all $\mathcal{Y}$ in B. The distance between two subspaces can be, e.g., defined by the projection 2-norm $\mathrm{dist}_{p2}(\mathcal{Y}, \mathcal{V}) = \|P_{\mathcal{Y}} - P_{\mathcal{V}}\|_2$, where $P_{\mathcal{Y}}$ and $P_{\mathcal{V}}$ denote the orthogonal projectors onto $\mathcal{Y}$ and $\mathcal{V}$, respectively [GV96]. Any compatible definition, such as the geodesic distance on the Grassmann manifold, can be used [EAS98].

Another Grassmannian generalization of the RQI, which uses a matrix shift instead of multiple scalar shifts, has been proposed in [AMSV02]. This iteration has been called Grassmann-RQI (GRQI).[1]

ALGORITHM 3.2 (GRQI). *Iterate the mapping* $\mathrm{Grass}(p,n) \ni \mathcal{Y} \mapsto \mathcal{Y}_+ \in \mathrm{Grass}(p,n)$ *defined by*

1. *Pick a basis* $Y \in \mathbb{R}^{n\times p}$ *that spans* $\mathcal{Y}$.
2. *Solve*

$$\boldsymbol{T}_Y Z := AZ - Z\underbrace{(Y^TY)^{-1}Y^TAY}_{R_A(Y)} = Y \tag{3.8}$$

for $Z \in \mathbb{R}^{n\times p}$.

3. *Define* $\mathcal{Y}_{k+1}$ *as the span of* Z.

The matrix $R_A(Y)$ can be interpreted as a block shift that reduces to the scalar Rayleigh quotient (3.2) in the case $p = 1$. The computations in Algorithm 3.2 are done in terms of $n \times p$ matrices, but they induce an iteration on the Grassmann manifold. That is, $\mathcal{Y}_{k+1}$ does not depend on the choice of the representative Y of $\mathcal{Y}_k$ chosen in (3.8). The GRQI method converges locally cubically to the spectral eigenspaces of A [Smi97, AMSV02].

Like the classical RQI mapping, which is ill-defined by (3.3) when $\rho(x)$ is an eigenvalue of A, the two iterations RSQR (Algorithm 3.1) and GRQI (Algorithm 3.2) are defined *almost* everywhere on $\mathrm{Grass}(p,n)$, i.e., there are points of singularity. In order to characterize these singularities, we introduce notations that will be used throughout the text. Let X denote an $n \times p$ orthonormal matrix (i.e., $X^TX = I$) that spans the current iterate, and let $[X|X_\perp]$ be an orthogonal $n \times n$ matrix. Define $A_{11} := X^TAX$, $A_{12} := X^TAX_\perp$, $A_{21} := X_\perp^TAX$, $A_{22} := X_\perp^TAX_\perp$. Let $\rho_1, \ldots, \rho_p$ denote the eigenvalues of A_{11} enumerated with their multiplicity. Then the RSQR and GRQI methods map the span of X to the span of an $n \times p$ matrix

$$X_+ = ZM, \tag{3.9}$$

where M is any invertible $p \times p$ matrix chosen so that $X_+^TX_+ = I$, and Z verifies

$$RSQR : (A - \rho_1 I)\ldots(A - \rho_p I)Z = X, \tag{3.10}$$

$$GRQI : AZ - ZA_{11} = X. \tag{3.11}$$

In RSQR, the matrices $(A - \rho_i I)$ are invertible if and only if the ρ_i's are not eigenvalues of A, in which case Z is well defined by the RSQR equation (3.10) and is full rank. In GRQI, a Sylvester equation (3.11) has to be solved. The solution Z exists and is unique if and only if the spectra of A and of A_{11} are disjoint. Indeed, rotating

[1] During the final preparation of this manuscript, the authors became aware of an independent derivation of the GRQI method in [Smi97].

$X \mapsto XQ$ so that $A_{11} = \mathrm{diag}(\rho_1, \ldots, \rho_p)$ decouples the Sylvester equation (3.11) into p linear systems of equations

$$(A - \rho_i)z_i = x_i,$$

where x_i and z_i denote the ith column of the rotated X and Z, respectively. So, the conditions for existence and uniqueness of Z are the same in both inverse iterations. An additional subtlety of GRQI is that the computed Z may a priori be rank deficient. However, numerical experiments suggest that if Z is the unique solution of the GRQI equation (3.11), then it is full rank (see [AH02] for details).

If the span of X is close to $\mathcal{V}$, then the eigenvalues of A_{11} are close to the eigenvalues of $A|_{\mathcal{V}}$, which are obviously eigenvalues of A. Therefore, (3.10) and (3.11) are intrinsically *ill-conditioned* when the span of X is close to an eigenspace $\mathcal{Y}$. This ill-conditioning is essential for the fast convergence of the shifted iterations and does *not* mean that the span of the computed Z is ill-conditioned as a function of X. This fact was already emphasized in the case $p = 1$ by Peters and Wilkinson [PW79]. The proof of cubic convergence of RSQR and GRQI shows that the span of Z is well conditioned when the span of X is "sufficiently close" to the target eigenspace $\mathcal{V}$. We shall see later (section 4.2) that the notion of "sufficiently close" depends on the structure of A.

3.2. Newton iterations. It comes as a direct consequence from the definitions in section 2 that the p-dimensional eigenbases of A are the full-rank $n \times p$ solutions of the matrix equation

$$F(Y) := \Pi_{Y_\perp} AY = 0, \tag{3.12}$$

where $\Pi_{Y_\perp} := I - Y(Y^TY)^{-1}Y^T$ is the orthogonal projector onto the orthogonal complement of $\mathrm{span}(Y)$. This formulation of eigenbasis computation as a zero finding problem calls for the utilization of the Newton iteration (see, e.g., [DS83, NW99]) in the Euclidean space $\mathbb{R}^{n\times p}$, which consists in solving the Newton equation

$$F(Y) + \mathrm{D}F(Y)[\Delta] = 0, \tag{3.13}$$

where $\mathrm{D}F(Y)[\Delta]$ denotes the directional derivative of F at Y in the direction of Δ, and performing the update

$$Y_+ = Y + \Delta. \tag{3.14}$$

However, the solutions of (3.12) are not isolated in $\mathbb{R}^{n\times p}$, namely, if Y is a solution, then all the elements of the equivalence class $Y\,\mathrm{GL}_p$ are solutions, too. In fact, since F is homogeneous of degree one, i.e., $F(YM) = F(Y)M$, the solution of the Newton equation (3.13), when unique, is $\Delta = -Y$. So any point Y is mapped to $Y_+ = 0$. This is clearly a solution of $F(Y) = 0$, but it spans the trivial zero-dimensional subspace.

A remedy consists in constraining Δ to belong to the *horizontal space*

$$H_Y := \{\Delta \in \mathbb{R}^{n\times p} : Y^T\Delta = 0\}, \tag{3.15}$$

orthogonal to the equivalence class $Y\,\mathrm{GL}_p$. With this constraint on Δ, the solutions Δ of $F(Y + \Delta) = 0$ become isolated. However, the Newton equation (3.13) has, in general, no solution Δ in H_Y, so the Newton equation (3.13) must be relaxed.

We will consider two approaches. The first one consists in projecting the Newton equation (3.13) onto H_Y

$$\Pi_{Y_\perp}(F(Y) + \mathrm{D}F(Y)[\Delta]) = 0,\ Y^T\Delta = 0. \tag{3.16}$$

The second approach consists in solving the Newton equation (3.13) in the least squares sense, that is,

$$\Delta = \arg\min_{Y^T\Delta=0} \|F(Y) + \mathrm{D}F(Y)[\Delta]\|^2. \tag{3.17}$$

In the remainder of the present section, we develop the ideas (3.16) and (3.17) and show how they relate to methods proposed in the literature.

Define a map $J_Y : H_Y \to H_Y$ by projecting the Fréchet derivative of F in a direction $\Delta \in H_Y$ back onto H_Y,

$$J_Y : H_Y \to H_Y : \Delta \mapsto \Pi_{Y_\perp}\mathrm{D}F(Y)[\Delta] = \Pi A\Pi\Delta - \Delta(Y^TY)^{-1}Y^TAY, \tag{3.18}$$

where Π is a shorthand notation for $\Pi_{Y_\perp}$. Using this notation, (3.16) may be written

$$J_Y[\Delta] = -F(Y). \tag{3.19}$$

The Newton–Grassman (NG) algorithm is formally stated as follows.

ALGORITHM 3.3 (NG). *Iterate the mapping* $\mathrm{Grass}(p,n) \ni \mathcal{Y} \mapsto \mathcal{Y}_+ \in \mathrm{Grass}(p,n)$ *defined by*

1. *Pick a basis* $Y \in \mathbb{R}^{n\times p}$ *that spans* $\mathcal{Y}$ *and solve the equation*

$$\Pi A\Pi\Delta - \Delta(Y^TY)^{-1}Y^TAY = -\Pi AY \tag{3.20}$$

under the constraint $Y^T\Delta = 0$, *where* $\Pi := I - Y(Y^TY)^{-1}Y^T$.

2. *Perform the update*

$$\mathcal{Y}_+ = \mathrm{span}(Y + \Delta). \tag{3.21}$$

One checks that $\mathcal{Y}_+$ does not depend on the Y chosen in step 1. Indeed, if Y yields the solution Δ of (3.20), then YM produces the solution ΔM for any $M \in \mathrm{GL}_p$, and $\mathrm{span}(Y + \Delta) = \mathrm{span}((Y + \Delta)M)$.

Algorithm NG admits the following geometric interpretation, valid for arbitrary A. The Grassmann manifold, endowed with the essentially unique Riemannian metric invariant by the action of the group of rotations, is a Riemannian manifold. In [Smi94], Smith proposes a Newton iteration on abstract Riemannian manifolds. This iteration, applied on the Grassmann manifold in order to solve (3.12), yields the search direction Δ given by (3.20), where Δ is interpreted as an element of the tangent space $T_Y\mathrm{Grass}(p,n)$; see [AMS02] for details. The update (3.21) is a simplification of the Riemannian updating procedure

$$\mathcal{Y}_+ = \mathrm{Exp}_{\mathcal{Y}}\Delta \tag{3.22}$$

consisting in following geodesics on the Grassmann manifold. Assuming $A = A^T$, Algorithm NG—but with geodesic update (3.22) instead of (3.21)—is also obtained by applying the Riemannian Newton method on $\mathrm{Grass}(p,n)$ for finding a stationary point of a generalized Rayleigh quotient [EAS98].

Algorithm NG was previously proposed for the case $A = A^T$ in [LST98], where quadratic convergence (at least) was proven. In [AMS02], it is shown that for

arbitrary A, NG with either geodesic update (3.22) or projected update (3.21) converges locally quadratically to the spectral p-dimensional eigenspaces of A. When $A = A^T$ (which is assumed to hold in the present paper) the rate of convergence of NG is shown to be *cubic*.

Now we turn to the least squares approach (3.17). As shown in the appendix, the solution Δ of the minimization problem (3.17) verifies

$$J^T \circ J[\Delta] + \Pi A Y (Y^T Y)^{-1} Y^T A^T \Pi \Delta = -J^T[F(Y)], \tag{3.23}$$

where J^T denotes the adjoint of the operator J (3.18) defined with respect to the inner product $\langle \Omega_1, \Omega_2 \rangle_X = \mathrm{trace}((X^T X)^{-1} \Omega_1^T \Omega_2)$. Assuming $A = A^T$, the operator J is self-adjoint and we obtain the following algorithm.

ALGORITHM 3.4 (NH). *Iterate the mapping* $\mathrm{Grass}(p, n) \ni \mathcal{Y} \mapsto \mathcal{Y}_+ \in \mathrm{Grass}(p, n)$ *defined by*

1. *Pick a basis* $Y \in \mathbb{R}^{n \times p}$ *that spans* $\mathcal{Y}$ *and solve the equation*

$$\begin{aligned} \Pi A^2 \Pi \Delta - 2\Pi A \Pi \Delta (Y^T Y)^{-1} Y^T A Y + \Delta (Y^T Y)^{-1} Y^T A Y (Y^T Y)^{-1} Y^T A Y \\ = -\Pi A \Pi A Y + \Pi A Y (Y^T Y)^{-1} Y^T A Y \end{aligned} \tag{3.24}$$

for the unknown Δ *under the constraint* $Y^T \Delta = 0$.

2. *Perform the update*

$$\mathcal{Y}_+ = \mathrm{span}(Y + \Delta). \tag{3.25}$$

Here again, it is checked that $\mathcal{Y}_+$ does not depend on the Y chosen in step 1. This least squares approach can be interpreted as a matrix generalization of the homogeneous Newton method proposed by Dedieu and Shub [DS00].

Algorithm NH converges locally cubically to the spectral eigenspaces of A. This property can be deduced from the corresponding property in NG. Applying the operator J on the NG equation (3.19) yields

$$J^T \circ J[\Delta] = -J^T[F(Y)] \tag{3.26}$$

which only differs from the NH equation (3.23) by the term $\Pi A Y (Y^T Y)^{-1} Y^T A^T \Pi \Delta$. Since $\Pi A Y$ is zero at the solution and smooth, the operators in the left-hand side of (3.26) and (3.19) differ only at the second order. Since the right-hand side is of first order, the discrepancy between the solutions Δ of the NH equation (3.24) and the NG equation (3.20) is cubic, whence cubic convergence of NG is preserved in NH.

Like the inverse iterations (section 3.1), the two Newton methods NG (Algorithm 3.3) and NH (Algorithm 3.4) have points of singularity. Let us rewrite the key equations in a slightly more compact form, using the notations of section 3.1. The two Newton iterations map the span of an orthonormal X to the span of

$$X_+ = (X + \Delta) M = (X + X_\perp H) M, \tag{3.27}$$

where M is chosen to orthonormalize X_+ (M can, e.g., be obtained by a QR factorization), and Δ or H verify

$$NG : \Pi A \Pi \Delta - \Delta A_{11} = -\Pi A X,\ X^T \Delta = 0, \tag{3.28}$$

$$\text{or } A_{22} H - H A_{11} = -A_{21}. \tag{3.29}$$

$$NH : \Pi A^2 \Pi \Delta - 2\Pi A \Pi \Delta A_{11} + \Delta A_{11}^2 = -\Pi A \Pi A X + \Pi A X A_{11},\ X^T \Delta = 0, \tag{3.30}$$

$$\text{or } (A_{21} A_{12} + A_{22} A_{22}) H - 2 A_{22} H A_{11} + H A_{11}^2 = -A_{22} A_{21} + A_{21} A_{11}. \tag{3.31}$$

The inverse iterations (RSQR and GRQI) and the Newton iterations (NG and NH) are built on very different principles. In the inverse iterations, a new basis Z appears directly as the solution of a linear system of equations that becomes more and more *ill-conditioned* (i.e., almost singular) as the iterate X approaches an eigenspace. In the Newton methods, a correction Δ, verifying the horizontality constraints, is computed and added to the current iterate X. It is thus not surprising that the two approaches involve different singularities. In NG (3.29), H exists and is unique if and only if the spectra of A_{22} and A_{11} are disjoint. Note the difference from inverse iterations: the matrix A is replaced by the *projected* matrix A_{22}. In NH (3.31), H exists and is unique if and only if the eigenvalues of the quadratic eigenvalue problem $(A_{21}A_{12} + A_{22}A_{22} - 2A_{22}\lambda + \lambda^2 I)x \equiv (A_{21}A_{12} + (A_{22} - \lambda I)^2)x = 0$ are distinct from the eigenvalues of A_{11}; see (4.6). When the span of X is close to $\mathcal{V}$, the residual matrix A_{21} has small norm, and the Sylvester operator on the left-hand side of (3.29) and (3.31) is *well-conditioned*. Indeed, the eigenvalues of A_{22} are close to those of $A|_{\mathcal{V}_\perp}$, the eigenvalues of A_{11} are close to those of $A_{\mathcal{V}}$, and the spectra of $A|_{\mathcal{V}_\perp}$ and $A_{\mathcal{V}}$ are separated since, by hypothesis, $\mathcal{V}$ is a spectral eigenspace.

4. Comparison of methods. In the previous section, we have formulated four iterations—two shifted inverse iterations (RSQR and GRQI) and two Newton methods (NG and NH)—that evolve on the Grassmann manifold of p-planes in $\mathbb{R}^n$ and converge locally cubically to the spectral p-dimensional eigenspaces of a symmetric $n \times n$ matrix A. Surprisingly, and in spite of different underlying approaches, RSQR and GRQI coincide with NG in the particular case $p = 1$, as pointed out by several authors [Shu86, Smi94, ADM$^+$02, MA03]. When $p > 1$, however, the four methods differ.

In the present section, we compare the iterations in terms of numerical cost and global behavior. Low numerical cost and large basins of attraction are two desirable features for methods that compute invariant subspaces from a first estimate.

4.1. Practical implementation. Comparing the implementation of the four different techniques depends to a large extent on the structure of the matrix A. If we assume first that A is dense, then all four methods have a comparable complexity, namely $O(pn^3)$, which mainly accounts for the p matrix factorizations that each requires. The RSQR solves

$$RSQR : (A - \rho_1 I) \cdots (A - \rho_p I)Z = X, \tag{4.1}$$

which involves p symmetric matrices $(A - \rho_i I)$. In the case of the three other methods, the first thing to do is to reduce A_{11} to a diagonal form. This is cheap since A_{11} is a $p \times p$ matrix and p is in practical applications typically much smaller than n. Moreover, the diagonalization always exists since A_{11} is symmetric. This diagonalization decouples (3.11), (3.28), or (3.30) into p independent systems of linear equations of the form

$$GRQI : (A - \rho_i I)z = x, \tag{4.2}$$

$$NG : \quad \Pi(A - \rho_i I)\Pi\delta = -\Pi A x, \ X^T\delta = 0 \tag{4.3}$$

$$\text{or } (A_{22} - \rho I)h = -A_{21}e, \tag{4.4}$$

$$NH : \quad \Pi(A - \rho_i I)^2\Pi\delta = -g, \ X^T\delta = 0 \tag{4.5}$$

$$\text{or } ((A_{21}A_{12} + A_{22}A_{22}) - 2\rho A_{22} + \rho^2 I)h = -(A_{22}A_{21} - A_{21}A_{11})e, \tag{4.6}$$

where $e \in \mathbb{R}^p$ is the eigenvector defined by $A_{11}e = \rho_i e$ and $x := Xe$, $z := Ze$, $\delta := \Delta e$, $h := He$.

Clearly $O(pn^3)$ seems excessive since most eigenvalue solvers require only $O(n^3)$ floating point operations (i.e., flops). A significant improvement is obtained by proceeding in three phases as follows: (i) reduce the matrix A to a tridiagonal form in $O(n^3)$ flops, (ii) compute an eigenspace of the tridiagonal matrix, (iii) compute the corresponding eigenspace of the original A in $O(n^2p)$ flops. We now focus on the second phase and assume that A is already in tridiagonal form. For RSQR the solution of (4.1) requires now $O(np^2)$ flops, while for GRQI (4.2) this is $O(np)$; the subsequent reorthogonalization of Z requires $O(np^2)$ for both methods. For the Newton updates NG and NH, we use an idea from [PW79] which shows that the direction of the solution z of (4.2) is also given by the direction of $x + \delta$ where

$$\begin{bmatrix} A - \rho_i I & x \\ x^T & 0 \end{bmatrix} \begin{bmatrix} \delta \\ m \end{bmatrix} = \begin{bmatrix} -Ax \\ 0 \end{bmatrix}.$$

In a similar fashion, one can rewrite the Newton methods NG and NH as $(n+p)\times(n+p)$ symmetric problems:

$$\begin{bmatrix} A - \rho_i I & X \\ X^T & 0 \end{bmatrix} \begin{bmatrix} \delta \\ m \end{bmatrix} = \begin{bmatrix} -Ax \\ 0 \end{bmatrix} \tag{4.7}$$

and

$$\begin{bmatrix} (A - \rho_i I)^2 & X \\ X^T & 0 \end{bmatrix} \begin{bmatrix} \delta \\ m \end{bmatrix} = \begin{bmatrix} -g \\ 0 \end{bmatrix}, \tag{4.8}$$

respectively, rather than solving the dense problems (4.4) and (4.6). When $(A - \rho_i I)$ is tridiagonal, (4.7) and (4.8) can be solved in $O(np^2)$ flops each. The LDL^T decomposition of $(A - \rho_i I)$ and the QR decomposition of $(A - \rho_i I)$ both require $O(n)$ flops. The above problems are then replaced by

$$\begin{bmatrix} LDL^T & X \\ X^T & 0 \end{bmatrix} \begin{bmatrix} \delta \\ m \end{bmatrix} = \begin{bmatrix} -Ax \\ 0 \end{bmatrix} \tag{4.9}$$

and

$$\begin{bmatrix} R^T R & X \\ X^T & 0 \end{bmatrix} \begin{bmatrix} \delta \\ m \end{bmatrix} = \begin{bmatrix} -g \\ 0 \end{bmatrix}, \tag{4.10}$$

respectively, where L has only two diagonals and R only three. Solving the systems (4.9) and (4.10) (possibly with iterative refinement to ensure stability) requires $O(np^2)$ flops each. For a tridiagonal matrix A, the complexity for all four methods is thus $O(np^2)$ in addition to the cost of phases (i) and (iii). We point out, however, that there exist very efficient numerical methods for computing all the eigenvectors of tridiagonal matrices such that the computed eigenvectors are orthogonal to working precision [DP03]. Moreover, the Multiple Relatively Robust Representations algorithm announced in [DP03] would compute p eigenvectors of a tridiagonal matrix with lower order of complexity, $O(np)$, than the one reported above.

If the matrix A is sparse or banded, say with bandwidth $2q+1$, then the numerical cost per iterate of GRQI, NG, or NH is $O(nq^2p) + O(np^2)$ assuming $p, q << n$. If the bandwidth is sufficiently narrow, namely, $q^2 \approx p$, then the numerical cost remains $O(np^2)$. For RSQR, assuming that the linear system (4.1) is solved by Gauss elimination and back-substitution, the numerical cost per iterate is $O(nq^2p)+O(nqp^2)$;

hence the complexity of the algorithm essentially increases by a factor q at most as long as $q \leq p$. Another possibility, explained in section 3.1, is to implement RSQR as a multishift QR algorithm [BD89]. Chasing a $p \times p$ bulge down a tridiagonal matrix can be done with approximately n Householder reflections of dimension $p \times p$ and applying those to X will yield the solution Z of (32). The numerical cost is thus $O(np^2)$, but this implicit method has to be implemented with care [BBM02a, BBM02b] in order to work properly.

Finally, if the matrix A is very large but sparse, one could consider alternative sparse matrix techniques such as reordering methods that reduce the bandwidth of A or even iterative methods. If an approximate solution is sought using an iterative solver, a stopping criterion has also to be chosen for the inner iteration. Computing the first iterates with high precision may be unnecessary [EW96]. Iterative solvers are considered for the case $p = 1$ in [SE02], including a comparison between the RQI equation (4.2) and the Newton equation (4.3).

4.2. Basins of attraction. The four subspace methods under investigation in this paper, i.e., the two inverse iterations RSQR (Algorithm 3.1) and GRQI (Algorithm 3.2) and the two Newton methods NG (Algorithm 3.3) and NH (Algorithm 3.4), display *local* cubic convergence to the spectral eigenspaces of the symmetric matrix A. By "local convergence," it is meant that around each p-dimensional eigenspace $\mathcal{V}$, there exists a ball B in the Grassmann manifold $\mathrm{Grass}(p, n)$ such that the iteration converges to $\mathcal{V}$ for all initial point in B. But nothing has been said yet about the size of these balls. This is, however, an important issue, since a large ball means that the iteration will converge to the target eigenspace even if the initial estimate is not very precise.

It has been shown for previously available methods that the basins of attraction are prone to deteriorate when some eigenvalues are clustered. Batterson and Smillie [BS89] have drawn the basins of attraction of the RQI for $n = 3$ and have shown that they deteriorate when two eigenvalues are clustered. The bounds involved in the convergence results of the methods analyzed in [Dem87] blow up when the external gap vanishes.

In the present section, we illustrate properties of the basins of attraction on three examples. The first two examples are low-dimensional problems ($n = 3$ and $p = 1, 2$) for which faithful two-dimensional pictures of the basins of attraction can be drawn (the dimension of $\mathrm{Grass}(1, 3)$ and $\mathrm{Grass}(2, 3)$ is two). The third example is a higher-dimensional case. In these examples, the matrices A are chosen to illustrate the influence of the eigenvalue gaps on the basins of attraction.

In order to graphically represent basins of attraction, we take advantage of the following facts. Let $\mathcal{F}_A$ denote one of the four iteration mappings mentioned above. The mappings are invariant by orthogonal changes of coordinates, i.e., $Q\mathcal{F}_A(\mathcal{Y}) = \mathcal{F}_{QAQ^T}(Q\mathcal{Y})$ for all Q orthogonal. Therefore, we work without loss of generality in a coordinate system in which A is diagonal. Moreover, once A is diagonal, the mappings are invariant by multiplication by a sign matrix. To show this, note that sign matrices are orthogonal, replace Q above by a sign matrix S and use the relation $SAS = A$. Consequently, it is sufficient to represent the basins of attraction in the first orthant. The other orthants are deduced by symmetry. Note also that the matrices A, $-A$, and $A - \sigma I$ yield the same sequences of iterates for all σ.

Example 1 (Dependence on external gap). We consider the case $n = 3$ and $p = 1$ (iterates are one-dimensional subspaces of $\mathbb{R}^3$). Then the two inverse iterations (RSQR and GRQI) reduce to the RQI, which is equivalent to NG (see, e.g., [Smi94]).

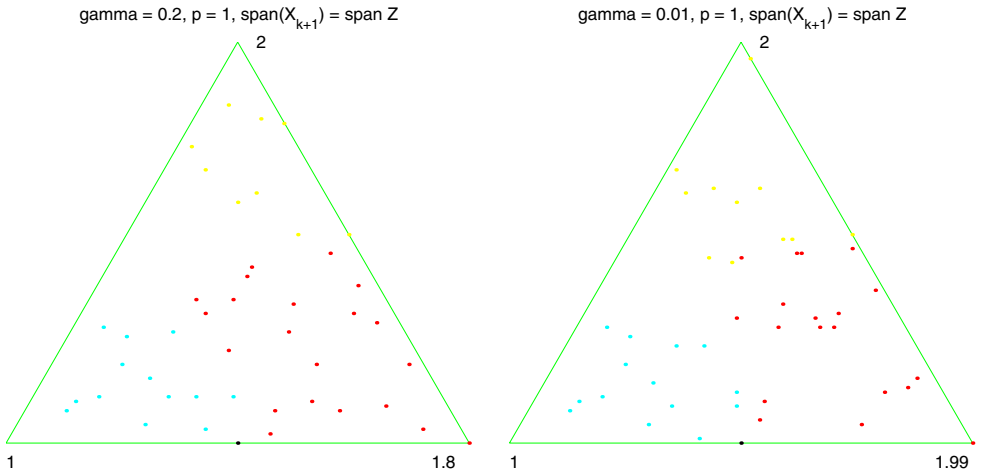


FIG. 4.1. *Basins of attraction for RSQR, GRQI, and NG ($n = 3$, $p = 1$). The three vertices correspond to the three eigenspaces. A similar figure appears in* [BS89]. *This figure also applies to NG with $n = 3$, $p = 2$ (see Example 2 in section 4.2).*

Figure 4.1 represents the basins of attraction of the RQI for $A = \mathrm{diag}(1, 2 - \gamma, 2)$. On the left-hand side of the figure $\gamma = .2$, and on the right-hand side γ is reduced to 0.01 in order to illustrate the effect of a small eigenvalue gap. Figure 4.1 should be read as follows. Displayed is the simplex $\{x \in \mathbb{R}^n : x_1 + x_2 + x_3 = 1, x_i > 0\}$. The iterates—one-dimensional subspaces of $\mathbb{R}^3$—are represented by their intersections with the simplex. The three vertices correspond to the three eigendirections of A, and the corresponding eigenvalues are indicated. The three colors indicate the three basins of attraction. It is seen that the basin of attraction of the upper vertex shrinks as its external gap is reduced. The basins of attraction of NH are qualitatively similar to the RQI-NG case. In conclusion, this simple example shows the dependence on external gap in all methods.

Example 2 (Dependence on internal gap in GRQI). We now investigate the case $n = 3$, $p = 2$ (iterates are 2-planes in $\mathbb{R}^3$) using the same two matrices A as above. Let us first consider the case of RSQR. Its basins of attraction are shown on Figure 4.2, where 2-planes are represented by the intersection of their normal vector with the simplex. The three vertices correspond to the three two-dimensional eigenspaces of A. For example, the upper vertex corresponds to the minor eigenspace. On the right-hand plot of Figure 4.2 and the ones that follow, the eigenspace represented by the lower left vertex has a small internal gap and a large external gap, while the two other vertices correspond to eigenspaces with a large internal gap and a small external gap. Figure 4.2 shows that the basins of attraction for RSQR collapse when the *external* gap is small. On this low-dimensional example, a small *internal* gap does not affect the basin of attraction.

The basins of attraction of GRQI are shown on Figure 4.3, with the same conventions as for the RSQR plot. One notices a peak growing towards the eigenspace with small internal gap. The tip of the peak is very close to the eigenspace, but this can hardly be seen on the figure because the peak is very narrow. This shows that for GRQI the basins of attraction may deteriorate around the eigenspaces with small internal gap. We will explain this feature analytically in section 4.3.

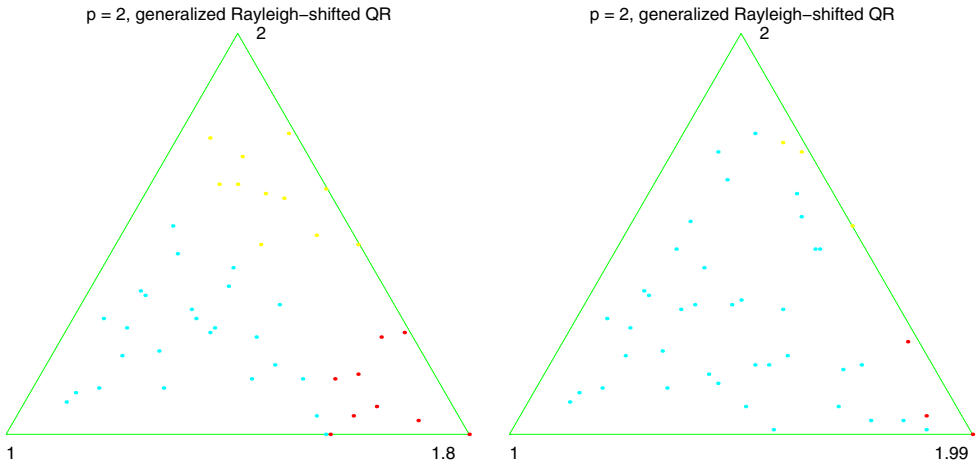


FIG. 4.2. *Basins of attraction for RSQR (case $p = 2$, $n = 3$). The elements of* Grass$(2, 3)$ *(i.e., 2-planes in $\mathbb{R}^3$) are represented by the intersection of their normal vector with the simplex.*

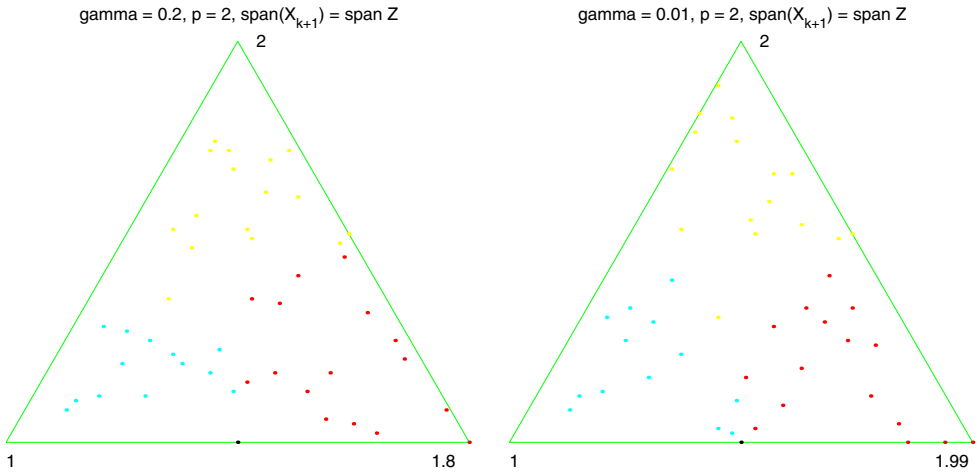


FIG. 4.3. *Basins of attraction for GRQI (case $p = 2$, $n = 3$).*

The Newton iteration NG displays the following *duality property*: If $\mathcal{X}^k$ is a sequence of iterates generated by NG, then $\mathcal{X}^k_\perp$ also forms a sequence of iterates of NG. To see this, let H verify the NG equation (3.29), note that $X_\perp - XH^T$ is orthogonal to $X + X_\perp H$, and $(-H^T)$ verifies $A_{11}(-H^T) - (-H^T)A_{22} = -A_{12}$, which is just the NG at the iterate $X_\perp$. By this duality property, the orthogonal complements of the iterates of NG ($p = 2$, $n = 3$) are one-dimensional iterates of NG ($p = 1$, $n = 3$). Representing 2-planes by the intersection of their normal vector with the simplex, the picture for NG in the case $p = 2$, $n = 3$ is the same as for the case $p = 1$, $n = 3$ illustrated on Figure 4.1.

The basins of attraction of the Newton iteration NH are shown in Figure 4.4, with the conventions explained above. The basins of attraction do not collapse in this low-dimensional example. One however should not conclude that everything goes well in higher dimensions, as we will show shortly.

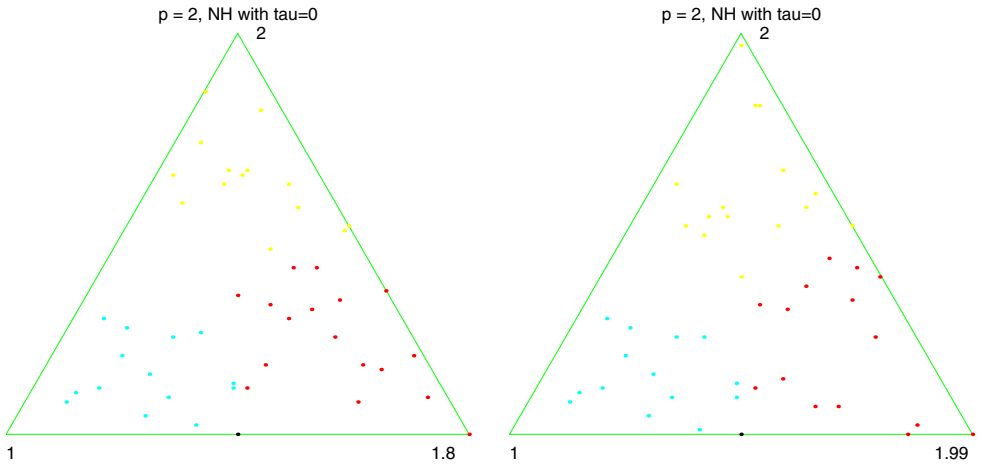


FIG. 4.4. *Basins of attraction for NH (case $p = 2$, $n = 3$). For the basins of attraction of NG ($p = 2$, $n = 3$), see Figure* 4.1.

In conclusion, this second example illustrates that the four methods are different when $p > 1$. It also reveals a dependence on internal gap occurring in GRQI.

Example 3 (Higher-dimensional case). The principal interest of the low-dimensional example studied above lies in the two-dimensional representation of the basins of attraction. We now consider an example in Grass$(3, 7)$, with dim Grass$(3, 7)$ $= 12$, in order to further investigate the influence of the eigenvalue gaps on the basins of attraction. We use the matrix

$$A = \mathrm{diag}(1, 2, 2 + \gamma, 2 + 2\gamma, 3, 4, 5),$$

where γ is a small number (we choose $\gamma = 10^{-2}$). We select three different eigenspaces in order to illustrate the influence of internal and external gaps. In each case, we pick 10^4 initial points randomly at three given distances of the targeted eigenspace and we count how often the sequence of iterates fails to converge to the target. We declare that the sequence converges if $\mathrm{dist}(\mathcal{X}^k, \mathcal{V}_{target}) < 10^{-6}$ with $k = 100$, where dist denotes the largest principal angle between the two arguments. The condition is usually already verified for very small k (see Figure 4.5), but if the iteration is started close to the boundary of the basin of attraction then the condition may be verified after arbitrarily many steps.

Here are the results of our experiments:

(i) Convergence to the eigenspace $\mathcal{V}_{leli}$ with eigenvalues 1, 3, and 4. This eigenspace has a large external gap and a large internal gap. The ratios of sequences that failed to converge to the targeted eigenspace are shown in Figure 4.5(a). As predicted by the theory, the four methods (RSQR, GRQI, NG, and NH) invariably converge to the targeted eigenspace when the initial error is small. When the initial error is large, the methods sometimes fail, and RSQR fails much more often than the other methods.

(ii) Convergence to $\mathcal{V}_{lesi}$ with eigenvalues 2, $2 + \gamma$, and $2 + 2\gamma$ (Figure 4.5(b)). This illustrates the influence of a small internal gap. All methods except GRQI have a large basin of attraction around $\mathcal{V}_{lesi}$. This confirms the information obtained in the

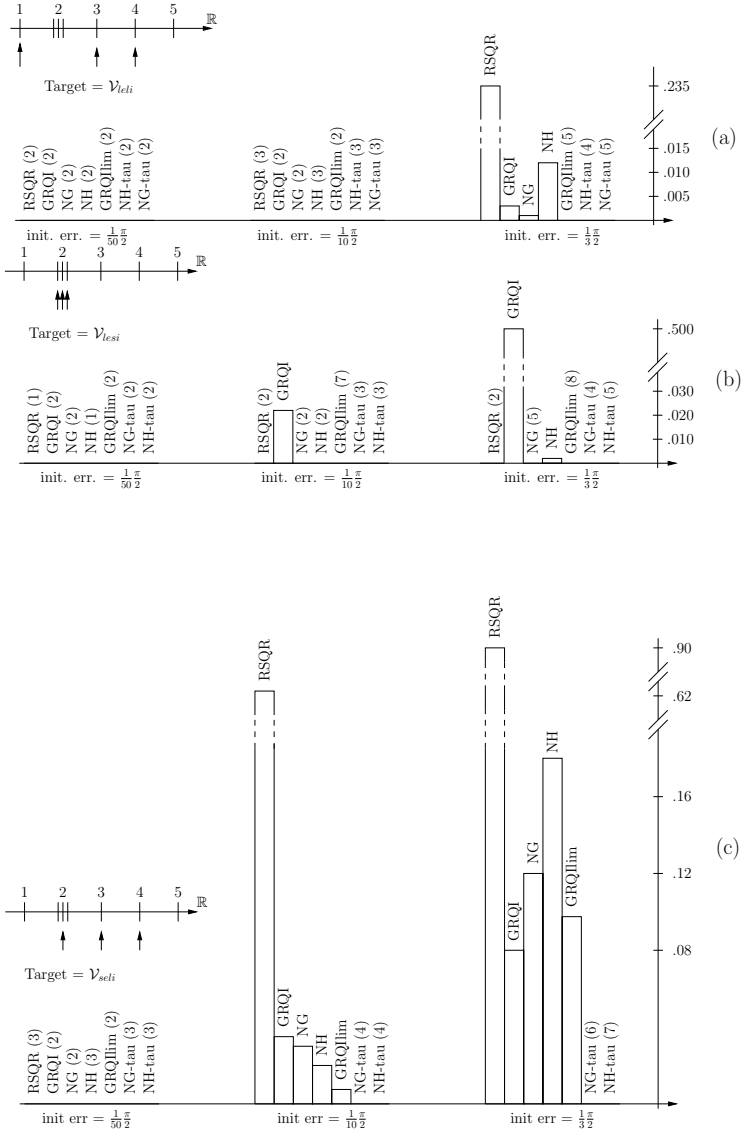


FIG. 4.5. *Ratio of sequences that failed to converge to the targeted eigenspace in Example* 3 *(section* 4*). "Init. err" gives the value of the largest principal angle between the initial subspace and the targeted eigenspace. Each ratio has been estimated using* 10^4 *randomly chosen starting points in each case. The absence of bar means that the sequence converged for all trials. We declare that a sequence converges if the largest principal angle between the* 100*th iterate and the target is smaller than* 10^{-6}*. The numbers between parentheses indicate the maximal number of iterates (evaluated on the* 10^4 *trials) necessary for the convergence condition to be satisfied.*

lower-dimensional case that the basins of attraction of eigenspaces with small internal gap are collapsed in GRQI (see the peak observed for GRQI in Figure 4.3).

In a different experiment not reported on the figure, we also considered initial

points situated at the distance $\frac{2}{3}\frac{\pi}{2}$ of $\mathcal{V}_{lesi}$. At such a distance, $\mathcal{V}_{lesi}$ is seldom the closest eigenspace, so convergence to $\mathcal{V}_{lesi}$ is not expected. And indeed, the iterates of GRQI, NG, and NH seldom converged to $\mathcal{V}_{lesi}$ (probability of convergence around 0.02). However, the iterates of RSQR did very often converge to $\mathcal{V}_{lesi}$, with probability 0.95. This means that the basin of attraction of $\mathcal{V}_{lesi}$ has a very large area under RSQR. It suggests that the eigenspaces with clustered eigenvalues have an oversized basin of attraction under RSQR, to the detriment of the other basins of attraction.

(iii) Convergence to $\mathcal{V}_{seli}$ with eigenvalues 2, 3, and 4 (Figure 4.5(c)). This eigenspace has a large internal gap but a small external gap. The number of failures of RSQR is about 10 times worse than for the other methods, and all the methods sometimes fail to converge to $\mathcal{V}_{seli}$ unless they are started very close to it. This means that the basin of attraction of $\mathcal{V}_{seli}$ is small for each method. Therefore, one usually tries to avoid small external gaps by enlarging the targeted eigenspace to include whole clusters of eigenvalues. However, this approach requires a priori information on the eigenvalues. In section 5 we will propose modified Newton methods that display large basins of attraction around eigenspaces like $\mathcal{V}_{seli}$.

4.3. Dependence on eigenvalue gaps. The numerical experiments reported in the previous section have led to the following observations. For the four methods under investigation, collapsed basins of attraction are observed around eigenspaces with small external eigenvalue gap. The basins of attraction of GRQI also deteriorate when the internal gap between eigenvalues is small. Under RSQR, the eigenspaces corresponding to clusters of eigenvalues have a particularly large basin of attraction. In the present section, we justify these observations analytically. As an aside, we obtain an alternative proof of cubic convergence for the Newton methods.

RSQR. For simplicity of the argument, consider $A = \mathrm{diag}(1, 1+\gamma, 2)$ with γ small. Let $\mathcal{V}$ be an eigenspace of A with small external gap, e.g., $\mathcal{V} = \mathrm{span}(e_2, e_3)$ corresponding to the eigenvalues $1+\gamma$ and 2. We now exhibit a subspace $\mathcal{X}^0$ close to $\mathcal{V}$ that is mapped by RSQR to a subspace close to $\mathrm{span}(e_1, e_2)$. Let $\mathcal{X}^0 = \mathrm{span}(e_2 + \alpha e_1, e_3 + \beta e_1)$ with $|\alpha|, |\beta| << 1$. Then $\mathcal{X}^0$ is close to $\mathcal{V}$. The Ritz values of $(A, \mathcal{X}^0)$ are $\rho_1 = 1 + \gamma - \alpha^2\gamma + O(\alpha^4) + O(\alpha^2\beta^2)$ and $\rho_2 = 2 - \beta^2 + O(\beta^4) + O(\alpha^2\beta^2)$, and one obtains for the new iterate computed by RSQR from $\mathcal{X}^0$

$$\mathcal{X}^1 = \mathrm{span}\left((A-\rho_1 I)^{-1}(A-\rho_2 I)^{-1} \begin{bmatrix} \alpha & \beta \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \simeq \mathrm{span} \begin{bmatrix} -\alpha^3 & 1 \\ 1 & 0 \\ 0 & \gamma/\beta^3 \end{bmatrix}.$$

If $\gamma << \beta^3$, then $\mathcal{X}^1$ is close to $\mathrm{span}(e_1, e_2)$. In other words, given a $\mathcal{X}^0$ that is close to $\mathcal{V}$ but does not contain e_3, if the cluster is sufficiently tight, then $\mathcal{X}^1$ is close to the eigenspace corresponding to the cluster. This shows that the basin of attraction of $\mathrm{span}(e_1, e_2)$ contains points close to $\mathcal{V}$.

The behavior we have just observed can be interpreted as a "cooperation" between clustered eigenvalues. If a Ritz value is a good shift for one eigenvalue in a cluster, it is also a good shift for all the eigenvalues in the cluster. Moreover, Ritz values of a randomly chosen subspace are more likely to be close to a cluster than to an isolated eigenvalue. This explains the oversized basins of attraction observed around eigenspaces with clustered eigenvalues.

GRQI. For GRQI, both a small external gap and a small internal gap may affect the quality of the basin of attraction of $\mathcal{V}$, as we now show.

GRQI maps the basis $Y = V + V_\perp K$ to the $Z = VZ_1 + V_\perp Z_2$, where

$$\Lambda_1 Z_1 + Z_1(I_p + K^T K)^{-1}(\Lambda_1 + K^T \Lambda_2 K) = I_p, \tag{4.11}$$

$$\Lambda_2 Z_2 + Z_2(I_p + K^T K)^{-1}(\Lambda_1 + K^T \Lambda_2 K) = K; \tag{4.12}$$

see [AMSV02, Abs03]. Define $K_+ = Z_2 Z_1^{-1}$ so that the span of Z is the same as the span of $V + V_\perp K_+$.

Let us first consider equation (4.12). This is a Sylvester equation. It is well-conditioned when K is small, therefore $Z_2 = O(K)$ due to the right-hand side. If the external gap of $\mathcal{V}$ is small, i.e., gap$[\Lambda_1, \Lambda_2]$ is small, then the Sylvester operator is arbitrarily ill-conditioned for small K, so Z_2 and K_+ may be large.

Now consider equation (4.11). This Sylvester equation is ill-conditioned when K is small. Since Λ_1 is diagonal, the lines of (4.11) are decoupled. Without loss of generality, let us consider the first line. Put $Z_1 = \left(\begin{smallmatrix}\zeta_{11} & \zeta_{12}\\ \zeta_{21} & \zeta_{22}\end{smallmatrix}\right)$, $\Lambda_1 = \left(\begin{smallmatrix}\sigma & \\ & \Sigma\end{smallmatrix}\right)$, $E = \Lambda_1 - (I_p + K^T K)^{-1}(\Lambda_1 + K^T \Lambda_2 K) = \left(\begin{smallmatrix}E_{11} & E_{12}\\ E_{21} & E_{22}\end{smallmatrix}\right)$. The first line of (4.11) yields (see [AMSV02])

$$\zeta_{11} = [E_{11} - E_{12}(\sigma I_p - \Sigma + E_{22})^{-1} E_{21}]^{-1}, \tag{4.13}$$

$$\zeta_{12} = -\zeta_{11} E_{12}(\sigma I_p - \Sigma + E_{22})^{-1}. \tag{4.14}$$

One obtains that $Z_1^{-1} = O(E) = O(K^2)$ and concludes that $K_+ = O(K^3)$, so the algorithm converges locally cubically [AMSV02]. However, if σ is close to an eigenvalue of Σ (i.e., if the internal gap is small), then $(\sigma I - \Sigma - E_{22})^{-1}$ is large for some small E (i.e., small K). This suggests that if the internal gap is small, there are some small Ks for which Z_1^{-1} is large, whence K_+ is large.

Newton methods. We show here that NG converges locally cubically to $\mathcal{V}$ and that the basin of attraction collapses when the external gap is small, but not when the internal gap is small. A similar development for NH leads to the same conclusions.

Let V be an orthonormal basis of the eigenspace $\mathcal{V}$ such that $V^T A V = \Lambda_1$ is diagonal, and let $V_\perp$ be an orthonormal basis of $\mathcal{V}_\perp$ such that $V_\perp^T A V_\perp = \Lambda_2$ is diagonal. The external gap of $\mathcal{V}$ is gap$[\Lambda_1, \Lambda_2]$. After some manipulations, one obtains that under NG (Algorithm 3.3 with projective update), the span of $V + V_\perp K$ is mapped to the span of $V + V_\perp K_+$, where K_+ verifies

$$K_+ = (K + (I + KK^T)^{-1}(L - K))(I - K^T(I + KK^T)^{-1}(L - K))^{-1}, \tag{4.15}$$

in which L solves

$$\begin{aligned}(\Lambda_2 + K\Lambda_1 K^T)(I + KK^T)^{-1} L - L\,(I + K^T K)^{-1}(\Lambda_1 + K^T \Lambda_2 K) \\ = [K\Lambda_1 K^T (I + KK^T)^{-1} + \Lambda_2((I + KK^T)^{-1} - I)]K \\ - K[(I + KK^T)^{-1} K^T \Lambda_2 K + ((I + K^T K)^{-1} - I)\Lambda_1].\end{aligned} \tag{4.16}$$

One deduces from (4.16) that $L = O(K^3)$ and then $K_+ = O(K^3)$, which means that the Newton iteration NG converges cubically; the reader is referred to [AMS02] for a detailed proof of cubic convergence. If the gap$[\Lambda_1, \Lambda_2]$ is small, then the Sylvester operator on the left-hand side of (4.16) becomes arbitrarily ill-conditioned for small K's (remember that the eigenvalues of a Sylvester operator are the differences between the eigenvalues of the two matrices involved in the equation [Ste73]), whence K_+ can be large even if K is small. This reasoning suggests that if the external gap of $\mathcal{V}$

is small, then some initial points close to $\mathcal{V}$ do not yield convergence to $\mathcal{V}$. On the other hand, the conditioning of the Sylvester operator in (4.16) is not affected by the internal gap of $\mathcal{V}$.

5. Improving the basins of attraction. Large basins of attraction are desirable as they ensure that the iteration will converge to the targeted eigenspace even if the initial subspace is a relatively poor estimate. The analysis in section 4 has shown that a small external gap, and in the case of GRQI a small internal gap, produces a degradation of the basins of attraction of the iterations defined in section 3. For this reason, we now discuss ways of improving the shape of the basins of attraction.

5.1. GRQI with limited variations. By experimenting with GRQI, we noticed that the sequences of iterates that diverge from the target eigenspace start with a big jump, i.e., the distance between the initial and second iterates is large. In an attempt to prevent this behavior, we apply a threshold value on the distance between two successive iterates.

This can be implemented in the following way. Let $\mathcal{X}$ be the current iterate and let X be an orthonormal $n \times p$ matrix that spans $\mathcal{X}$. Let θ_{max} be a threshold value on the principal angles between $\mathcal{X}$ and $\mathcal{X}_+$. Compute Z, the solution of the GRQI equation (3.11). Orthonormalize Z, e.g., by a Gram–Schmidt process. Then, by the CS decomposition theorem [PW94, GV96], there exist orthogonal matrices U_1 and V_1 and an orthonormal matrix Y with $Y^T X = 0$ such that

$$ZV_1 = XU_1C + YS,$$

where $C = \operatorname{diag}(\cos(\theta_1), \ldots, \cos(\theta_p))$, $S = \operatorname{diag}(\sin(\theta_1), \ldots, \sin(\theta_p))$, with $0 \leq \theta_1 \leq \cdots \leq \theta_p \leq \frac{\pi}{2}$. The θ_i's are the principal angles between $\operatorname{span}(X)$ and $\operatorname{span}(Z)$, and the columns of XU_1 and ZV_1 are the corresponding principal vectors. Define $\theta_i^{new} = \min\{\theta_i, \theta_{max}\}$. Then define $C^{new} = \operatorname{diag}(\cos(\theta_1^{new}), \ldots, \cos(\theta_p^{new}))$, $S^{new} = \operatorname{diag}(\sin(\theta_1^{new}), \ldots, \sin(\theta_p^{new}))$, and let the new iterate $\mathcal{X}_+$ be the span of $Z^{new} = XU_1C^{new} + YS^{new}$.

The matrix Z^{new} is obtained from the original Z in $O(np^2)$ flops by computing the singular value decomposition $X^T Z = U_1 C V_1^T$, then $S = \sin(\arccos C)$, and solving $YS = ZV_1 - XU_1C$. In fact, only the last columns of U_1, S, and Y corresponding to the θ_i's larger than the threshold θ_{max} have to be computed; the other columns of ZV_1 are unmodified in Z^{new}.

We chose $\theta_{max} = \frac{\pi}{10}$ in numerical experiments. The basins of attraction of this modified GRQI are displayed on Figure 5.1 for the low-dimensional case ($n = 3$, $p = 2$) investigated in the previous section (Example 2). Compare Figure 4.3 (GRQI) and Figure 5.1: the peak has been removed. Experimental results for the higher-dimensional case (Example 3 in the previous section) are displayed on Figure 4.5 (see columns labelled "GRQIlim"). They illustrate that this heuristic effectively suppresses the problem of dependence on the internal gap.

Another (arguably more natural) modification of GRQI consists in taking $\mathcal{X}_+$ on the Grassmann geodesic [EAS98, AMS02] between $\mathcal{X}$ and $\operatorname{span}(Z)$ with $\theta_p(\mathcal{X}, \mathcal{X}_+) = \theta_{max}$. This amounts to defining $\theta_i^{new} = \lambda\theta_i$ with $\lambda = \frac{\theta_{max}}{\theta_p}$. However, the previously described technique works slightly better in experiments.

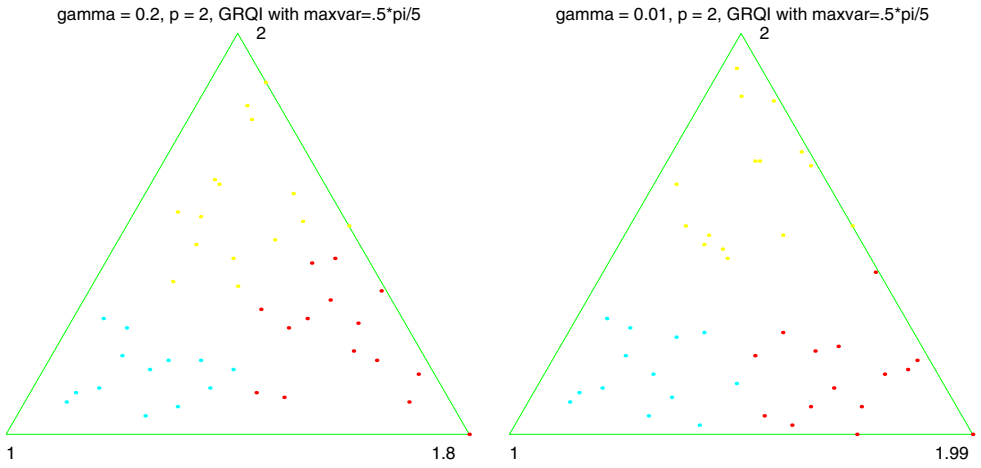


FIG. 5.1. *Basins of attraction for GRQI with limited steps (section* 5.1*) in the case* $p = 2$, $n = 3$. *Compare with the original GRQI (Figure* 4.3*).*

5.2. Modified Newton methods.

Deformation parameter τ. As explained in section 3.2, the NG iteration attempts to find a p-plane $\mathcal{Y}$ such that each basis Y of $\mathcal{Y}$ verifies

$$F(Y) := \Pi_{Y_\perp} AY = 0, \tag{5.1}$$

where $\Pi_{Y_\perp} := I - Y(Y^TY)^{-1}Y^T$. Equation (5.1) holds if and only if $\mathcal{Y}$ is an invariant subspace of A.

Let us define a *cost function*

$$f(Y) := \frac{1}{2}\mathrm{trace}((Y^TY)^{-1}F(Y)^TF(Y)). \tag{5.2}$$

It is easily checked that $f(Y)$ depends only on the span of Y, and not on the basis Y itself [AMS02]. So, the cost function f defines a scalar field on the Grassmann manifold. This scalar field is zero at the eigenspaces of A and strictly positive everywhere else. An illustration of the level curves of f is shown on Figure 5.2. We stress that f reaches its minimum value (zero) at all the eigenspaces of A, and not only at an extremal eigenspace. This is a fundamental difference with the more familiar Rayleigh quotient.

Section 4 has shown that the basins of attraction of the two Newton methods (NG and NH) deteriorate in the presence of a small external gap. On the other hand, Figure 5.2 suggests that the basins of attraction of the steepest descent flow of the cost function f remain broad even when the eigenvalue gap shrinks. A numerical simulation of the steepest descent flow of f in Example 3 of section 4 shows that the distance between each eigenspace and the boundary of its basin of attraction is large (greater than $\frac{1}{3}\frac{\pi}{2}$) in all cases.

This prompts us to follow the steepest descent of f when the solution is far away from a solution and use the Newton method in the neighborhood of a solution. It is, however, difficult to decide when the commutation between the two behaviors should occur. If the Newton iteration takes over too soon, the basins of attraction may be collapsed. If the transition occurs late in the iterative process, then more steps will be

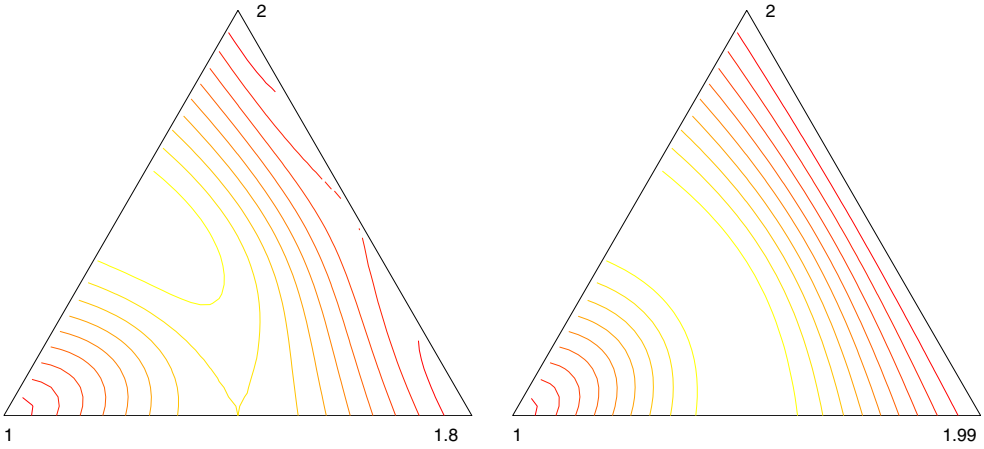


FIG. 5.2. *Level curves of the cost function f defined in* (5.2). *The cost function vanishes at the three eigenspaces (represented by the three vertices) and is strictly positive everywhere else. The gradient descent flow for f consists in following the steepest descent path of these level curves.*

necessary before obtaining a good approximation of the eigenspace. A remedy is to implement a smooth transition between the two behaviors by means of a deformation parameter, an idea which connects with trust region methods (see, e.g., [DS83] or Chap. 11 in [NW99]). We now show how this deformation approach works out in our case.

Let Y be a basis for the current subspace, let H_Y be the horizontal space defined as in (3.15), and let $J : H_Y \to H_Y : \Pi \mathrm{D}F(Y)[\Delta]$ be as in (3.18). The derivative of the cost function f in the direction of Δ, with $Y^T\Delta = 0$, is given by

$$\begin{aligned} Df(Y)[\Delta] &= \operatorname{trace}((Y^TY)^{-1}F(Y)^T DF(Y)[\Delta]) \\ &= \operatorname{trace}((Y^TY)^{-1}F(Y)^T J[\Delta]) \\ &= \operatorname{trace}((Y^TY)^{-1}(J^T[F(Y)])^T\Delta) \\ &= \operatorname{trace}(\Delta^T J^T[F(Y)](Y^TY)^{-1}), \end{aligned}$$

where J^T denotes the adjoint of the operator J defined with respect to the inner product $\langle \Omega_1, \Omega_2 \rangle_Y = \operatorname{trace}((Y^TY)^{-1}\Omega_1^T\Omega_2)$. Then a formula in [AMS02] directly yields

$$\operatorname{grad} f(Y) = J^T[F(Y)]. \tag{5.3}$$

On the other hand, the NG equation (3.20) reads $J[\Delta] = -F(Y)$, or equivalently

$$J^T \circ J[\Delta] = -J^T[F(Y)].$$

A continuous deformation between the gradient descent flow of f and the Newton method NG is thus given by

$$(J^T \circ J + \tau\,\mathrm{Id})[\Delta] = -J^T[F(Y)].$$

If τ is small, then Δ is close to the NG-vector given by the NG equation (3.28), and the iteration is close to pure NG. If τ is large, then the direction of Δ is close to the negative gradient of f, and the iteration is similar to a Euler integration of the

gradient descent flow of f. Because we assume $A = A^T$, the operator J is self-adjoint and the modified NG algorithm can be expressed as follows.

ALGORITHM 5.1 (NG-tau). *Iterate the mapping $\mathcal{Y} \mapsto \mathcal{Y}_+$ defined by*

1. *Pick an orthonormal basis Y that spans $\mathcal{Y}$ and solve the equation*

(5.4)
$$\Pi A \Pi A \Pi \Delta + \Delta Y^T A Y Y^T A Y - 2\Pi A \Pi \Delta Y^T A Y + \tau \Delta = -(\Pi A \Pi A Y - \Pi A Y Y^T A Y),$$

where $\Pi := (I - YY^T)$, under the constraint $Y^T \Delta = 0$.

2. *Perform the update $\mathcal{Y}_+ = \mathrm{span}(Y + \Delta)$.*

We now introduce a τ deformation parameter in the NH iteration such that the limiting cases $\tau = 0$ and $\tau = \infty$ correspond to pure NH and gradient descent for f, respectively. This is easily done because the right-hand side of the NH equation (3.30) is precisely $-\operatorname{grad} f$ (compare (3.23) and (5.3)).

ALGORITHM 5.2 (NH-tau). *Iterate the mapping $\mathcal{Y} \mapsto \mathcal{Y}_+$ defined by*

1. *Pick an orthonormal basis Y that spans $\mathcal{Y}$ and solve the equation*

(5.5)
$$\Pi A^2 \Pi \Delta + \Delta Y^T A Y Y^T A Y - 2\Pi A \Pi \Delta Y^T A Y + \tau \Delta = -(\Pi A \Pi A Y - \Pi A Y Y^T A Y),$$

where $\Pi := (I - YY^T)$, under the constraint $Y^T \Delta = 0$.

2. *Perform the update $\mathcal{Y}_+ = \mathrm{span}(Y + \Delta)$.*

Note that the only difference between NG-tau and NH-tau is in the first term of (5.4) and (5.5).

Practical implementation. The major computational work in NG-tau (Algorithm 5.1) or NH-tau (Algorithm 5.2) is solving (5.4) or (5.5) for Δ. Like in the case of the original NG and NH iterations (see section 4.1), the first thing to do is to diagonalize the small $p \times p$ matrix $A_{11} := Y^T A Y$. This decouples (5.4) or (5.5) into p individual systems of linear equations of the form

$$((\Pi A \Pi - \rho_i I)^2 - \tau I)\delta = -g, \ Y^T \delta = 0, \tag{5.6}$$

$$\Pi((A - \rho_i I)^2 - \tau I)\Pi \delta = -g, \ Y^T \delta = 0 \tag{5.7}$$

for NG-tau and NH-tau, respectively. In the case of NH-tau (5.7), the projectors are outside the matrix, which allows for the utilization of the techniques described in section 4.1. It is possible to obtain the Cholesky decomposition of $(A-\rho_i)^2 - \tau I = R_\tau^T R_\tau$ from that of $(A-\rho_i)^2 = R^T R$ in $O(n)$ flops when R has only three diagonals [Par80]. The algorithm NH-tau will thus again require $O(np^2)$ flops per iteration. In the case of NG-tau, in the absence of an efficient algorithm for solving (5.6), the cost for producing a new iterate involves $O(n^3)$ flops, even if A is tridiagonal. Thus, NH-tau has a serious advantage over NG-tau in terms of numerical cost.

Choosing the deformation parameter. There exist many strategies for tuning the τ parameter in order to improve the global behavior of the algorithm while preserving the ultimate rate of convergence of the Newton method. In a line search approach, one selects τ so that the direction of K remains in a sector around the negative gradient of f and then perform a line search along the direction of the K computed from (5.4). Equation (5.4) is also helpful in trust region methods. A large τ corresponds to a small trust region, while $\tau = 0$ corresponds to a trust region that contains the exact next Newton iterate.

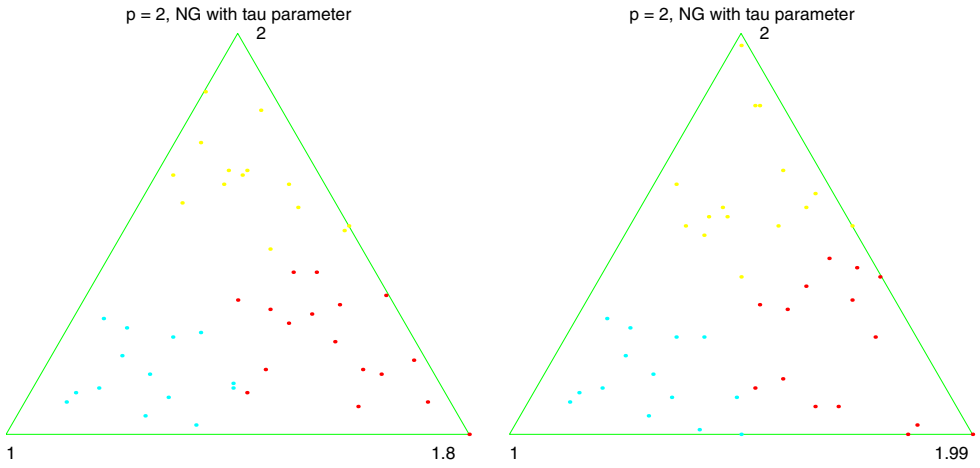


FIG. 5.3. *Attraction basins for NG with* $\tau := f$ (5.4) *in the case* $p = 2$, $n = 3$. *Compare with original NG on Figure* 4.1. *Local cubic convergence is preserved.*

Classical strategies for choosing τ involve several parameters that the user can choose at his convenience [DS83, NW99]. In the present case, the very simple choice $\tau := f$ preserves the local cubic convergence and considerably enlarges the basins of attraction around the eigenspaces, both for NG-tau and NH-tau.

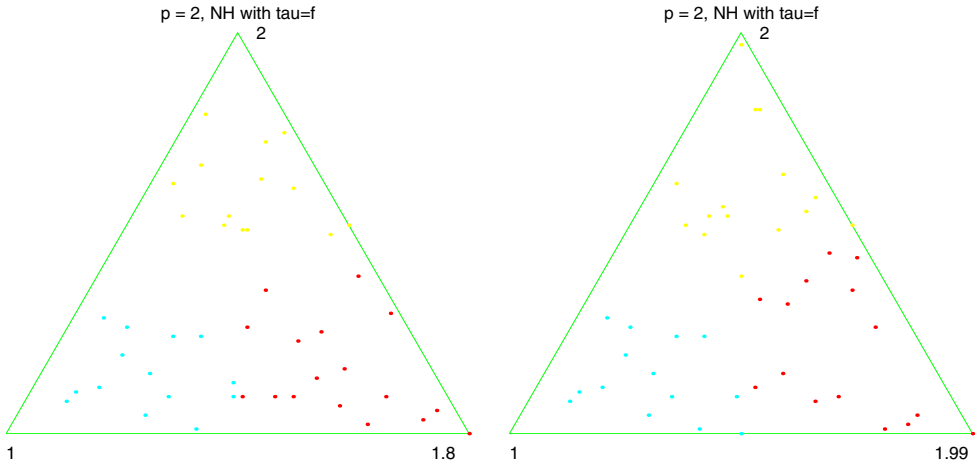


FIG. 5.4. *Attraction basins for NH-tau with* $\tau := f$ (5.4) *in the case* $p = 2$, $n = 3$.

Local cubic convergence of NG-tau and NH-tau with $\tau = f$ is direct: τ is quadratic in the distance between the current iterate $\mathcal{Y}$ and the target eigenspace $\mathcal{V}$, while the right-hand side of (5.4) or (5.5) is linear in the distance. Consequently, the perturbation on the solution K of (5.4) induced by $\tau = f$ is cubic.

The global behavior of NG-tau and NH-tau is illustrated on Figures 5.3 and 5.4 in our low-dimensional example utilized in section 4.2. Comparison with Figure 4.1 shows that the basins of attraction have been considerably enlarged around the three eigenspaces. The improvement is even more spectacular in the larger dimensional case (Example 3 in section 4.2); see Figure 4.5. Both NG-tau and NH-tau invari-

ably converged to the targeted eigenspace. We had to choose the largest principal angle between the first iterate and the target greater than $\frac{1}{2.2}\frac{\pi}{2}$ in order to observe convergence to a wrong eigenspace.

Note that the balls centered on the eigenspaces of A overlap if their radius (measured in arc length on Grassmann [AMS02] or by means of the largest principal angle) is larger than $\pi/4$. So there is a geometrical limitation on the size of the basins of attraction. Our results show that in the NG-tau and NH-tau, the basins of attraction are so large that the geometrical limit is almost reached.

6. Conclusion. We have compared four iterative methods, i.e., RSQR (Algorithm 3.1), GRQI (Algorithm 3.2), NG (Algorithm 3.3), and NH (Algorithm 3.4), that operate on the set of p-dimensional subspaces of $\mathbb{R}^n$ and refine initial estimates of invariant subspaces of a symmetric $n \times n$ matrix A with cubic rate of convergence. Methods RSQR and GRQI are formulated as shifted inverse iterations. The former uses multiple scalar shifts while the latter involves a matrix shift. Algorithms NG and NH are derived from a Newton argument. Algorithm RSQR can be traced back to [PK69, PP73] and its proof of cubic local convergence is implicitly contained in [WE91]. GRQI is studied in [Smi97, AMSV02]. NG appears in [LST98, EAS98, LE02] and is connected to [Ste73, DMW83, Cha84, Dem87, Fat98, DF01]. Its local rate of convergence is studied in [AMS02]; see also sections 3.2 and 4.3. To our knowledge, NH was never mentioned before in the literature.

We have shown that although these four iterations converge locally cubically to the spectral (i.e., isolated) eigenspaces of A, they appreciably differ in their global behavior. The basin of attraction of an eigenspace $\mathcal{V}$ collapses when the eigenvalues of A relative to $\mathcal{V}$ are not well separated from the other eigenvalues of A. Moreover, in the case of GRQI, the basin of attraction of $\mathcal{V}$ also deteriorates if the eigenvalues relative to $\mathcal{V}$ are clustered. This dependence on eigenvalue gaps means that the sequence of iterates may diverge from $\mathcal{V}$ even if the initial point is a good approximation of $\mathcal{V}$.

For three of the methods, we have proposed ways of improving the shape of the basins of attraction. In the GRQI case, our numerical experiments suggest that a simple heuristic imposing a limitation on the distance between successive iterates removes the bad influence of clustered eigenvalues in the target eigenspace. In the Newton case, we have introduced a deformation parameter that achieves a continuous deformation between the pure Newton case (NG or NH) and the gradient descent flow of a cost function. Our experiments show that a simple choice of the deformation parameter spectacularly improves the shape of the basins of attraction while preserving the ultimate cubic convergence rate.

We also commented on the practical implementation of the various iterations. With the exception of the deformed NG iteration, a new iterate of each method can be computed in $O(np^2)$ flops when A has bandwidth $2q + 1$ and $q = O(p^{1/2})$. When $q = 1$ there exist very efficient methods that compute all eigenvectors; see [DP03]. When A is sparse but not banded the computational cost of one iteration step will depend on the type of sparsity, but the complexity is essentially that of p sparse solves and therefore likely to be only linear in n.

In the Newton methods presented here, it is essential to compute the updates with high accuracy in order to take advantage of the cubic rate of convergence. Another approach consists in using acceleration techniques that exploit the useful information given by the previous updates in order to improve the current approximate solution. This allows for lower accuracy solves of the Newton equations, e.g., using iterative solvers; see [FSV98, Kny01] for more details. In the $p = 1$ case, this approach yields

e.g., the celebrated Jacobi–Davidson method [SV96] for which the use of iterative solvers as inner solution process is well understood [Not02, Not03]. As an aside, the Jacobi–Davidson method is equivalent to RQI with $p = 1$ when the Newton equations are solved exactly (this rejoins our remark on the $p = 1$ case in section 4). In the $p > 1$ case, we obtain a "block Jacobi–Davidson" that was touched upon in recent references [LE02, Bra03].

Among the algorithms considered here, our study suggests the NH algorithm with deformation parameter (Algorithm 5.2) as the method of choice for its remarkable combination of advantages: excellent global behavior, cubic rate of convergence, and low numerical cost $O(np^2)$ when A is suitably condensed.

Appendix. Derivation of Algorithm NH. In this section, we explain how the NH equation, i.e., (3.23) or (3.24), is derived from the minimization problem (3.17).

Let F be defined as in (3.12), $F(Y) := \Pi_{Y_\perp} AY$ and let H_Y denote the horizontal space (3.15), $H_Y := \{Y^T\Delta = 0\}$. Let $\mathbb{J}$ denote the operator $DF(Y)$ restricted to act on H_Y,

$$\mathbb{J}[\Delta] = \Pi A\Pi\Delta - \Delta(Y^TY)^{-1}Y^TAY - Y(Y^TY)^{-1}\Delta^TAY = J[\Delta] - Y(Y^TY)^{-1}\Delta^TAY,$$

where J denotes the operator $\Pi DF(Y)$ defined in (3.18) restricted to act on H_Y. Let

$$m_Y(\Delta) := \frac{1}{2}\|F(Y) + \mathbb{J}[\Delta]\|^2 = \frac{1}{2}\mathrm{trace}((Y^TY)^{-1}(F(Y) + \mathbb{J}[\Delta])^T(F(Y) + \mathbb{J}[\Delta])),$$

where the $(Y^TY)^{-1}$ factor is introduced so that $m_{YM}(\Delta M) = m_Y(\Delta)$ for all $M \in \mathrm{GL}_p$ (this allows us to take Y not necessarily orthonormal).

The minimization problem (3.17) is to compute $\Delta^* = \arg\min_{\Delta\in H_Y} m_Y(\Delta)$. To this end, define $\mathbb{J}^T$, the adjoint of $\mathbb{J}$, by requiring that $\mathbb{J}^T$ is on $\mathbb{R}^{n\times p}$ into H_Y and verifies $\mathrm{trace}((Y^TY)^{-1}\Omega^T\mathbb{J}[\Delta]) = \mathrm{trace}((Y^TY)^{-1}(\mathbb{J}^T[\Omega])^T\Delta)$ for all $\Omega \in \mathbb{R}^{n\times p}$ and all $\Delta \in H_Y$. One obtains

$$\mathbb{J}^T[\Omega] = J^T[\Pi\Omega] - \Pi AY(Y^TY)^{-1}\Omega^TY$$

and

$$J^T : H_Y \to H_Y : \Delta \mapsto \Pi A^T\Pi\Delta - \Delta(Y^TY)^{-1}YA^TY^T.$$

Then one readily obtains

$$\mathrm{D}m_Y(\Delta)[\tilde{\Delta}] = \mathrm{trace}((Y^TY)^{-1}(\mathbb{J}^T[F(Y)] + \mathbb{J}^T \circ \mathbb{J}[\Delta])^T\tilde{\Delta});$$

hence the solution Δ^* of the minimization problem (3.17) verifies the normal equations $\mathbb{J}^T \circ \mathbb{J}[\Delta^*] = -\mathbb{J}^T[F(Y)]$ that is

$$J^T \circ J[\Delta^*] + \Pi AY(Y^TY)^{-1}Y^TA^T\Delta^* = -J[F(Y)]. \tag{6.1}$$

If $A = A^T$, then J is self-adjoint and the latter equation develops into the NH equation (3.24).

Acknowledgments. We thank M. Petre for his contribution to the Matlab script that generates pictures of basins of attraction.

This paper presents research partially supported by the Belgian Programme on Inter-university Poles of Attraction, initiated by the Belgian State, Prime Minister's

Office for Science, Technology and Culture. Part of this work was performed while the first author was a guest at the Mathematisches Institut der Universität Würzburg under a grant from the European Nonlinear Control Network. The hospitality of the members of the department is gratefully acknowledged. The work was completed while the first and second authors were visiting the department of Mechanical and Aerospace Engineering at Princeton University. The hospitality of the members of the department, especially Prof. N. Leonard, is gratefully acknowledged. The second author thanks N. Leonard and E. Sontag for partial financial support under U.S. Air Force grants F49620-01-1-0063 and F49620-01-1-0382.

REFERENCES

[Abs03] P.-A. Absil, *Invariant Subspace Computation: A Geometric Approach*, Ph.D. thesis, Faculté des Sciences Appliquées, Université de Liège, Liège, Belgium, 2003.

[ADM+02] R. L. Adler, J.-P. Dedieu, J. Y. Margulies, M. Martens, and M. Shub, *Newton's method on Riemannian manifolds and a geometric model for the human spine*, IMA J. Numer. Anal., 22 (2002), pp. 359–390.

[AH02] P.-A. Absil, U. Helmke and K. Hüper, *Well-posedness and regularity properties of the Grassman-Rayleigh quotient iteration*, Found. Comput. Math., submitted.

[AMS02] P.-A. Absil, R. Mahony, and R. Sepulchre, *Riemannian geometry of Grassmann manifolds with a view on algorithmic computation*, Acta Appl. Math., 80 (2004), pp. 199–220 (manuscript available from http://www.montefiore.ulg.ac.be/systems/Publi/Grass_geom.htm).

[AMSV02] P.-A. Absil, R. Mahony, R. Sepulchre, and P. Van Dooren, *A Grassmann-Rayleigh quotient iteration for computing invariant subspaces*, SIAM Rev., 44 (2002), pp. 57–73.

[BBM02a] 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.

[BBM02b] 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.

[BD89] Z. Bai and J. Demmel, *On a block implementation of Hessenberg QR iteration*, Intl. J. High Speed Comput., 1 (1989), pp. 97–112; also available online as LAPACK Working Note 8 from http://www.netlib.org/lapack/lawns/lawn08.ps and http://www.netlib.org/lapack/lawnspdf/lawn08.pdf.

[Bra03] J. Brandts, *The Riccati algorithm for eigenvalues and invariant subspaces of matrices with inexpensive action*, Linear Algebra Appl., 358 (2003), pp. 335–365.

[BS89] S. Batterson and J. Smillie, *The dynamics of Rayleigh quotient iteration*, SIAM J. Numer. Anal., 26 (1989), pp. 624–636.

[Cha84] F. Chatelin, *Simultaneous Newton's iteration for the eigenproblem*, Comput. Suppl., 5 (1984), pp. 67–74.

[Dem87] J. W. Demmel, *Three methods for refining estimates of invariant subspaces*, Computing, 38 (1987), pp. 43–57.

[DF01] L. Dieci and M. J. Friedman, *Continuation of invariant subspaces*, Numer. Linear Algebra Appl., 8 (2001), pp. 317–327.

[DMW83] J. J. Dongarra, C. B. Moler, and J. H. Wilkinson, *Improving the accuracy of computed eigenvalues and eigenvectors*, SIAM J. Numer. Anal., 20 (1983), pp. 23–45.

[DP03] I. S. Dhillon and B. N. Parlett, *Orthogonal eigenvectors and relative gaps*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 858–899.

[DS83] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall Ser. Comput. Math., Prentice Hall, Englewood Cliffs, NJ, 1983.

[DS00] J.-P. Dedieu and M. Shub, *Multihomogeneous Newton method*, Math. Comp., 69 (2000), pp. 1071–1098.

[EAS98] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.

[EW96] S. Eisenstat and H. Walker, *Choosing the forcing terms in an inexact Newton method*, SIAM J. Sci. Comput., 17 (1996), pp. 16–32.

[Fat98] J.-L. Fattebert, *A block Rayleigh quotient iteration with local quadratic convergence*, Electron. Trans. Numer. Anal., 7 (1998), pp. 56–74.

[FGP94] J. Ferrer, M. I. García, and F. Puerta, *Differentiable families of subspaces*, Linear Algebra Appl., 199 (1994), pp. 229–252.

[FSV98] D. R. Fokkema, G. L. G. Sleijpen, and H. A. Van der Vorst, *Accelerated inexact Newton schemes for large systems of nonlinear equations*, SIAM J. Sci. Comput., 19 (1998), pp. 657–674.

[GV96] G. H. Golub and C. F. Van Loan, *Matrix Computations,* 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[HM94] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems*, Springer-Verlag, London, 1994.

[Ips97] I. C. F. Ipsen, *Computing an eigenvector with inverse iteration*, SIAM Rev., 39 (1997), pp. 254–291.

[JS01] Z. Jia and G. W. Stewart, *An analysis of the Rayleigh-Ritz method for approximating eigenspaces*, Math. Comp., 70 (2001), pp. 637–647.

[Kny01] A. V. Knyazev, *Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.

[LE02] E. Lundström and L. Eldén, *Adaptive eigenvalue computations using Newton's method on the Grassmann manifold*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 819–839.

[LST98] R. Lösche, H. Schwetlick, and G. Timmerman, *A modified block Newton iteration for approximating an invariant subspace of a symmetric matrix*, Linear Algebra Appl., 275/276 (1998), pp. 381–400.

[MA03] R. Mahony and P.-A. Absil, *The continuous-time Rayleigh quotient flow on the sphere*, Linear Algebra Appl., 368 (2003), pp. 343–357.

[Not02] Y. Notay, *Combination of Jacobi-Davidson and conjugate gradients for the partial symmetric eigenproblem*, Numer. Linear Algebra Appl., 9 (2002), pp. 21–44.

[Not03] Y. Notay, *Convergence analysis of inexact Rayleigh quotient iteration*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 627–644.

[NW99] J. Nocedal and S. Wright, *Numerical optimization*, Springer Ser. Oper. Res., Springer-Verlag, New York, 1999.

[Par80] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1980; republished by SIAM, Philadelphia, 1998.

[PK69] B. N. Parlett and W. Kahan, *On the convergence of a practical QR algorithm*, in Information Processing 68 (Proc. IFIP Congress, Edinburgh, 1968), Vol. 1: Mathematics, Software, North-Holland, Amsterdam, 1969, pp. 114–118.

[PP73] B. N. Parlett and W. G. Poole, *A geometric theory for the QR, LU and Power Iteration*, SIAM J. Numer. Anal., 10 (1973), pp. 389–412.

[PS95] R. D. Pantazis and D. B. Szyld, *Regions of convergence of the Rayleigh quotient iteration method*, Numer. Linear Algebra Appl., 2 (1995), pp. 251–269.

[PW79] G. Peters and J. H. Wilkinson, *Inverse iteration, ill-conditioned equations and Newton's method*, SIAM Rev., 21 (1979), pp. 339–360.

[PW94] C. C. Paige and M. Wei, *History and generality of the CS decomposition*, Linear Algebra Appl., 208/209 (1994), pp. 303–326.

[RR02] A. C. M. Ran and L. Rodman, *A class of robustness problems in matrix analysis*, in Interpolation Theory, Systems Theory and Related Topics, The Harry Dym Anniversary Volume, D. Alpay, I. Gohberg, and V. Vinnikov, eds., Oper. Theory Adv. Appl. 134, Birkhäuser, Basel, 2002, pp. 337–383.

[SE02] V. Simoncini and L. Eldén, *Inexact Rayleigh quotient-type methods for eigenvalue computations*, BIT, 42 (2002), pp. 159–182.

[Shu86] M. Shub, *Some remarks on dynamical systems and numerical analysis*, in Proceedings VII ELAM, L. Lara-Carrero and J. Lewowicz, eds., Equinoccio, U. Simón Bolívar, Caracas, Venezuela, 1986, pp. 69–91.

[Smi94] S. T. Smith, *Optimization techniques on Riemannian manifolds*, in Hamiltonian and Gradient Flows, Algorithms and Control, A. Bloch, ed., Fields Inst. Commun. 3, AMS, Providence, RI, 1994, pp. 113–136.

[Smi97] P. Smit, *Numerical Analysis of Eigenvalue Algorithms Based on Subspace Iterations*, Ph.D. thesis, CentER, Tilburg University, Tilburg, The Netherlands, 1997.

[Ste73] G. W. Stewart, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Rev., 15 (1973), pp. 727–764.

[SV96] 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.
[Wat82] D. S. Watkins, *Understanding the QR algorithm*, SIAM Rev., 24 (1982), pp. 427–440.
[WE91] D. S. Watkins and L. Elsner, *Convergence of algorithms of decomposition type for the eigenvalue problem*, Linear Algebra Appl., 143 (1991), pp. 19–47.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 1, pp. 97–124

SPLITTING A MATRIX OF LAURENT POLYNOMIALS WITH SYMMETRY AND ITS APPLICATION TO SYMMETRIC FRAMELET FILTER BANKS*

BIN HAN† AND QUN MO†

Abstract. Let M be a 2×2 matrix of Laurent polynomials with real coefficients and symmetry. In this paper, we obtain a necessary and sufficient condition for the existence of four Laurent polynomials (or finite-impulse-response filters) u_1, u_2, v_1, v_2 with real coefficients and symmetry such that

$$\begin{bmatrix} u_1(z) & v_1(z) \\ u_2(z) & v_2(z) \end{bmatrix} \begin{bmatrix} u_1(1/z) & u_2(1/z) \\ v_1(1/z) & v_2(1/z) \end{bmatrix} = M(z) \qquad \forall\, z \in \mathbb{C}\backslash\{0\}$$

and $[Su_1](z)[Sv_2](z) = [Su_2](z)[Sv_1](z)$, where $[Sp](z) = p(z)/p(1/z)$ for a nonzero Laurent polynomial p. Our criterion can be easily checked and a step-by-step algorithm will be given to construct the symmetric filters u_1, u_2, v_1, v_2. As an application of this result to symmetric framelet filter banks, we present a necessary and sufficient condition for the construction of a symmetric multiresolution analysis tight wavelet frame with two compactly supported generators derived from a given symmetric refinable function. Once such a necessary and sufficient condition is satisfied, an algorithm will be used to construct a symmetric framelet filter bank with two high-pass filters which is of interest in applications such as signal denoising and image processing. As an illustration of our results and algorithms in this paper, we give several examples of symmetric framelet filter banks with two high-pass filters which have good vanishing moments and are derived from various symmetric low-pass filters including some B-spline filters.

Key words. matrix splitting, symmetry, framelet filter banks, tight wavelet frames, low-pass and high-pass filters, refinable functions

AMS subject classifications. 15A23, 15A54, 42C40

DOI. 10.1137/S0895479802418859

1. Introduction and motivation.

Matrix theory plays an important role in wavelet analysis [4] and filter banks [17, 18]. In this paper, we are interested in splitting a 2×2 matrix of Laurent polynomials with real coefficients and symmetry into the form $U(z)U(1/z)^T$ for some 2×2 matrix U whose entries are Laurent polynomials with real coefficients and symmetry. Our investigation on this matrix splitting problem is greatly motivated by the recent development of symmetric tight wavelet frames and framelet filter banks which have been found to be useful and interesting in many applications [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. In the following, let us review some necessary background and explain our motivation to study this problem.

Since Daubechies constructed her famous family of compactly supported orthonormal wavelet bases in 1988, wavelets have been extensively studied and successfully applied to many areas. Though orthonormal wavelet bases have many desired properties in applications, as Daubechies pointed out in [4], except the Haar wavelet which is discontinuous, there is no compactly supported real-valued continuous orthonormal wavelet basis that can have symmetry. However, in many applications, for

*Received by the editors November 28, 2002; accepted for publication (in revised form) by A. H. Sayed August 26, 2003; published electronically August 6, 2004. This research was supported by the Natural Science and Engineering Research Council of Canada (NSERC Canada) under grant G121210654.

http://www.siam.org/journals/simax/26-1/41885.html

†Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, AB, Canada T6G 2G1 (bhan@math.ualberta.ca, http://www.ualberta.ca/∼bhanqmo @math.ualberta.ca, http://www.math.ualberta.ca/∼qmo).

various purposes, symmetry is a much desired property. In order to achieve symmetry in a wavelet system or a wavelet filter bank, many generalizations of orthonormal wavelet bases have been proposed and investigated in the literature [4, 18]. In this paper, we are particularly interested in tight wavelet frames and framelet filter banks which currently stimulate a lot of interest in both theory and application due to their particular interesting features [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. As a generalization of an orthonormal wavelet basis, a tight wavelet frame is an overcomplete wavelet system that preserves many desirable properties of an orthonormal wavelet basis. See Selesnick [15] for discussion on applications and interesting features of tight wavelet frames and framelet filter banks.

Before proceeding further, let us review some definitions and notation. We say that a set $\{\psi^1,\dots,\psi^r\}$ of functions in $L_2(\mathbb{R})$ generates a (normalized) *tight wavelet frame* in $L_2(\mathbb{R})$ if

$$\|f\|^2=\sum_{\ell=1}^{r}\sum_{j\in\mathbb{Z}}\sum_{k\in\mathbb{Z}}|\langle f,\psi^\ell_{j,k}\rangle|^2 \qquad \forall\, f\in L_2(\mathbb{R}) \quad \text{with} \quad \psi^\ell_{j,k}:=2^{j/2}\psi^\ell(2^j\cdot-k), \tag{1.1}$$

where $\langle f,g\rangle:=\int_{\mathbb{R}} f(x)\overline{g(x)}\,dx$ and $\|f\|^2:=\langle f,f\rangle$. The set $\{\psi^1,\dots,\psi^r\}$ is called a set of generators for the corresponding tight wavelet frame. Let δ denote the *Dirac sequence* such that $\delta_0=1$ and $\delta_k=0$ for all $k\in\mathbb{Z}\backslash\{0\}$. In particular, if $\{\psi^1,\dots,\psi^r\}$ generates a tight wavelet frame and $\langle\psi^\ell_{j,k},\psi^{\ell'}_{j',k'}\rangle=\delta_{\ell-\ell'}\delta_{j-j'}\delta_{k-k'}$ for all $\ell,\ell'=1,\dots,r$ and $j,j',k,k'\in\mathbb{Z}$, then $\{\psi^1,\dots,\psi^r\}$ generates an orthonormal wavelet basis in $L_2(\mathbb{R})$. It follows directly from (1.1) that any function $f\in L_2(\mathbb{R})$ has the wavelet expansion: $f=\sum_{\ell=1}^r\sum_{j\in\mathbb{Z}}\sum_{k\in\mathbb{Z}}\langle f,\psi^\ell_{j,k}\rangle\psi^\ell_{j,k}$.

In order to have a fast algorithm, one is interested in tight wavelet frames which are derived from refinable functions via the multiresolution analysis (MRA). We say that a function ϕ is *refinable* if $\hat\phi(2\xi)=a(e^{-i\xi})\hat\phi(\xi)$ for a Laurent polynomial a with $a(1)=1$ (a is called the mask for the refinable function ϕ and is also called a low-pass filter in engineering), where the Fourier transform is defined to be $\hat f(\xi)=\int_{\mathbb{R}} f(x)e^{-i\xi x}\,dx$ for $f\in L_1(\mathbb{R})$ and can be naturally extended to tempered distributions. We usually normalize a refinable function ϕ by $\hat\phi(0)=1$.

Throughout this paper, we assume that all Laurent polynomials have real coefficients. In other words, all the filters discussed in this paper are of finite-impulse-response (FIR) and have real coefficients.

As an important family of refinable functions, B-spline functions are useful in applications. The B-spline function of order n ($n\in\mathbb{N}$), denoted by B_n throughout this paper, can be obtained via the recursive formula: $B_1:=\chi_{[0,1]}$, the characteristic function of the interval $[0,1]$, and $B_n(x):=\int_0^1 B_{n-1}(x-t)\,dt$ for $n\geqslant 2$. The B-spline function $B_n\in C^{n-2}(\mathbb{R})$ is a symmetric refinable function satisfying $\widehat{B_n}(2\xi)=2^{-n}(1+e^{-i\xi})^n\widehat{B_n}(\xi)$ for $\xi\in\mathbb{R}$.

In order to obtain an orthonormal wavelet basis from a refinable function ϕ via the MRA, the refinable function ϕ must satisfy the following condition [4, 18]:

$$\int_{\mathbb{R}}\phi(x+k)\overline{\phi(x)}\,dx=\delta_k \qquad \forall\, k\in\mathbb{Z}. \tag{1.2}$$

By a simple argument, (1.2) implies that its mask a must satisfy the following condi-

tion [4, 18]:

$$|a(z)|^2 + |a(-z)|^2 = 1 \qquad \forall\, z \in \mathbb{T} := \{z \in \mathbb{C} \,:\, |z| = 1\}. \tag{1.3}$$

If (1.2) holds, one can define a wavelet function by $\hat{\psi}(2\xi) = e^{-i\xi} a(-e^{i\xi})\hat{\phi}(\xi)$. Then $\{\psi\}$ generates an orthonormal wavelet basis in $L_2(\mathbb{R})$ (see [4]). Note that the Haar wavelet is derived from the B-spline function B_1 which is discontinuous.

The conditions in (1.2) and (1.3) impose a very restrict constraint on a refinable function and its low-pass filter. Many refinable functions such as the B-spline functions $B_n (n > 1)$ do not satisfy (1.2). In fact, up to an integer shift, B_1 is the only example of real-valued compactly supported refinable function that can have symmetry and satisfy (1.2) (see [4]).

As discussed above, an orthonormal wavelet basis has only one generator. By increasing the number of generators in a tight wavelet frame, recently it was found that one has a lot of freedom in the construction of tight wavelet frames derived from refinable functions which may not satisfy the condition in (1.2). For example, it was demonstrated in Ron and Shen [14] that from any B-spline function of order n, one can construct a symmetric tight wavelet frame with n generators. More recently, Chui and He [1] (also see Petukhov [12]) showed that if the mask a for a symmetric refinable function satisfies

$$|a(z)|^2 + |a(-z)|^2 \leqslant 1 \qquad \forall\, z \in \mathbb{T}, \tag{1.4}$$

then one can derive a symmetric tight wavelet frame with three generators. Recently, Daubechies et al. [6] and Chui, He, and Stöckler [2] obtained the following interesting procedure that yields all possible MRA tight wavelet frames derived from a refinable function.

THEOREM 1.1. *Let ϕ be a refinable function in $L_2(\mathbb{R})$ such that $\hat{\phi}(2\xi)=a(e^{-i\xi})\hat{\phi}(\xi)$ for a Laurent polynomial a with $a(1) = 1$. Suppose that there exist Laurent polynomials $\Theta, a^1, \ldots, a^r$ such that $\Theta(1) = 1$ and*

$$\begin{bmatrix} a^1(z) & \cdots & a^r(z) \\ a^1(-z) & \cdots & a^r(-z) \end{bmatrix} \begin{bmatrix} a^1(1/z) & a^1(-1/z) \\ \vdots & \vdots \\ a^r(1/z) & a^r(-1/z) \end{bmatrix} = M_\Theta(z), \tag{1.5}$$

where

$$M_\Theta(z) := \begin{bmatrix} \Theta(z) - \Theta(z^2)a(z)a(1/z) & -\Theta(z^2)a(z)a(-1/z) \\ -\Theta(z^2)a(-z)a(1/z) & \Theta(-z) - \Theta(z^2)a(-z)a(-1/z) \end{bmatrix}, \quad z \in \mathbb{C}\backslash\{0\}. \tag{1.6}$$

Define the wavelet functions $\psi^1, \ldots, \psi^r$ by $\widehat{\psi^\ell}(2\xi) = a^\ell(e^{-i\xi})\hat{\phi}(\xi)$, $\ell = 1, \ldots, r$. Then $\{\psi^1, \ldots, \psi^r\}$ generates a tight wavelet frame in $L_2(\mathbb{R})$.

According to Theorem 1.1, a framelet filter bank consists of a low-pass filter a and r high-pass filters $a^1, \ldots, a^r$. In order to design a framelet filter bank, one has to split the matrix M_Θ in (1.6) into the form of (1.5).

Using Theorem 1.1, it was demonstrated in [2] (also cf. [6]) that for any refinable function $\phi \in L_2(\mathbb{R})$ whose integer shifts are stable, one can obtain an MRA tight wavelet frame with two generators. Unfortunately, when ϕ is symmetric, the construction in [2, 6] cannot guarantee the symmetry of the two constructed generators which do not have symmetry in most cases.

Though by increasing the number of generators in a tight wavelet frame one has a great deal of freedom to construct them from refinable functions, in many applications, for various purposes such as computational cost and storage concern, one prefers a symmetric tight wavelet frame with a small as possible number of generators (or equivalently, high-pass filters). Ideally, a tight wavelet frame with a single symmetric generator is desirable. However, as shown in [2, 6], it is impossible to have an MRA symmetric tight wavelet frame with one continuous generator. All the above discussions naturally motivate us to consider construction of symmetric MRA tight wavelet frames with two generators (that is, symmetric framelet filter banks with two high-pass filters) for the following possible advantages.

(1) Such framelet filter banks have symmetry which is a much desired property in applications.

(2) By using two high-pass filters, one still has much freedom to construct symmetric framelet filter banks from many low-pass filters without imposing strict conditions on them.

(3) By limiting to two high-pass filters, the associated framelet transform for decomposition and reconstruction is efficient in terms of computational and storage costs.

(4) Such symmetric framelet filter banks can have good vanishing moments, short support and many other desired properties.

In order to construct a symmetric framelet filter bank with two high-pass filters, according to Theorem 1.1, the core problem is to find two symmetric high-pass filters a^1 and a^2 such that (1.5) holds with $r = 2$. In other words, we have to split the 2×2 matrix M_Θ of Laurent polynomials into the desirable form in (1.5). This motivates us to investigate the problem of splitting a matrix of Laurent polynomials with symmetry which may be of interest in other applications such as construction of symmetric orthonormal multiwavelets and dual framelet filter banks [2, 5, 6].

The following is an outline of this paper. In section 2, we shall present a general result on splitting a matrix of Laurent polynomials with symmetry. As an application of this result to symmetric framelet filter banks, we shall present a necessary and sufficient condition for the construction of a symmetric tight wavelet frame with two generators derived from a given symmetric refinable function through Theorem 1.1. Once the necessary and sufficient condition is satisfied, we shall present a step-by-step algorithm (see Algorithm 2.5 in section 2) to derive the two symmetric high-pass filters from a given low-pass filter. In section 3, we shall present some examples of symmetric framelet filter banks with two high-pass filters which are derived from various low-pass filters including some B-spline filters. Our work in this paper was also motivated by [11, 13, 15], where symmetric tight wavelet frames with two generators were considered but using the unitary extension principle in [14], which is a special case of Theorem 1.1 by taking $\Theta = 1$. In this paper, we shall generalize [13] by considering the general fundamental function Θ instead of the special case $\Theta = 1$. As discussed in [2, 6], a nonconstant Θ is very important in order to have a tight wavelet frame with good vanishing moments. Also, in order to use the unitary extension principle, the mask must satisfy (1.4) which excludes some interesting low-pass filters [1, 2, 6, 11, 12]. We shall see that by using the general construction in Theorem 1.1 the investigation of symmetric tight wavelet frames and symmetric framelet filter banks becomes much more complicated. This paper is also motivated by [9], which proves that one can derive from any B-spline function of order m ($m \in \mathbb{N}$) an MRA tight wavelet frame in $L_2(\mathbb{R})$ which is generated by the dyadic dilates and integer shifts

of three compactly supported real-valued symmetric wavelet functions with vanishing moments of the highest possible order m. For multivariate tight wavelet frames, see Han [7] and references therein.

In section 3, by using Algorithm 2.5 and Theorem 1.1 we shall give examples to show that symmetric framelet filter banks with two high-pass filters having good vanishing moments can be constructed. For applications of framelet filter banks, see [15]. In order to prove the main results in this paper, in section 4, we shall provide some auxiliary results. In section 5, we shall prove our main result on splitting a matrix of Laurent polynomials with symmetry. Though the whole proof of the main result is somewhat technical, we shall present a step-by-step algorithm (see Algorithm 5.1 in section 5) to implement the main result on splitting a matrix of Laurent polynomials with symmetry which may be of interest in other applications.

2. Main results. In this section, we shall present the main results of this paper. We shall obtain a general result on splitting a matrix of Laurent polynomials with symmetry. As an application of such a result, we shall give a necessary and sufficient condition for the construction of symmetric MRA tight wavelet frames with two compactly supported generators. A step-by-step algorithm (Algorithm 2.5) will be given for construction of symmetric framelet filter banks.

In order to state the results in this section, let us introduce some notation first. We remind the reader that all of the Laurent polynomials discussed in this paper have real coefficients and we say that a Laurent polynomial p with real coefficients is symmetric (or antisymmetric) about $k/2$ for some $k \in \mathbb{Z}$ if $p(z) = z^k p(1/z)$ (or $p(z) = -z^k p(1/z)$). Throughout this paper, we say that a Laurent polynomial p is *(anti)symmetric* if p is either symmetric or antisymmetric. For a nonzero Laurent polynomial p, we define an operator S to be

$$[Sp](z) := \frac{p(z)}{p(1/z)}, \qquad z \in \mathbb{C}\backslash\{0\}. \tag{2.1}$$

When $p \equiv 0$, by convention Sp is undefined and can be anything.

The following result can be easily verified.

PROPOSITION 2.1. *Let p and q be two Laurent polynomials with real coefficients. Then*

(1) *p is (anti)symmetric about $k/2$ for some $k \in \mathbb{Z}$ if and only if $[Sp](z) = \pm z^k$.*

(2) *$[S(p(1/\cdot))](z) = [Sp](1/z) = 1/[Sp](z)$.*

(3) *$[S(pq)](z) = [Sp](z)[Sq](z)$ and $[S((\cdot)^k)](z) = z^{2k}$ for $k \in \mathbb{Z}$.*

(4) *If p and q are (anti)symmetric such that $Sp = Sq$, then $p \pm q$ is (anti)symmetric and $S(p \pm q) = Sp = Sq$.*

For a nonzero Laurent polynomial $p(z) = \sum_{k=\ell}^{h} p_k z^k$ such that $p_\ell \neq 0$ and $p_h \neq 0$, we denote the *degree* of p by $\deg(p) = h - \ell$. In other words, $\deg(p)$ measures the length of the filter p. By convention, $\deg(0) = -\infty$. For any two Laurent polynomials p and q, we say that $p \mid q$ if there is another Laurent polynomial h such that $q(z) = p(z)h(z)$ for all $z \in \mathbb{C}\backslash\{0\}$. We define $\gcd(p, q)$ to be a nonzero Laurent polynomial h with maximum degree such that $h \mid p$ and $h \mid q$. By convention, $\gcd(0, 0) = 0$. We say that a Laurent polynomial p is *trivial* if $p(z) = cz^k$ for some $c \in \mathbb{R}\backslash\{0\}$ and $k \in \mathbb{Z}$. Up to a factor of a trivial Laurent polynomial, $\gcd(p, q)$ is unique.

In the terminology of digital signal processing, the symmetries of filters are classified into type I to type IV filters according to whether the filter is symmetric or antisymmetric with an even or odd degree. The operator S defined in (2.1) is very

useful in this paper to distinguish these four types of symmetries of filters. See Table 1 for more detail.

TABLE 1
Type I *to type* IV *symmetries of a filter* p *described in terms of the operator* S *defined in* (2.1). *In this table,* k *is an integer and even (or odd) means the filter* p *has an even (or odd) degree.*

Type I	Type II	Type III	Type IV
symmetric/odd	symmetric/even	antisymmetric/odd	antisymmetric/even
$[Sp](z) = z^{2k+1}$	$[Sp](z) = z^{2k}$	$[Sp](z) = -z^{2k+1}$	$[Sp](z) = -z^{2k}$

PROPOSITION 2.2. *Let* $A(z) = A_0 + \sum_{k=1}^{N} A_k(z^{-k}+z^k)$ *with* $A_N \neq 0$ *be a Laurent polynomial with real coefficients. Then* $A(z) = d(z)d(1/z)$ *for some (anti)symmetric Laurent polynomial* d *with real coefficients if and only if* $A(z) = d_A(z)d_A(1/z)$ *for the Laurent polynomial* d_A *which is uniquely determined by one of the following four cases:*

Case 1. *When* $N = 2n$ *and* $A_N > 0$, *define* $d_A(z) = c_0 + \sum_{k=1}^{n} c_k(z^k + z^{-k})$ *and* $sgn(A_N) = 1$.

Case 2. *When* $N = 2n$ *and* $A_N < 0$, *define* $d_A(z) = \sum_{k=0}^{n} c_k(z^k - z^{-k})$ *and* $sgn(A_N) = -1$.

Case 3. *When* $N = 2n+1$ *and* $A_N > 0$, *define* $d_A(z) = \sum_{k=0}^{n} c_k(z^k + z^{-1-k})$*and* $sgn(A_N) = 1$.

Case 4. *When* $N = 2n+1$ *and* $A_N < 0$, *define* $d_A(z) = \sum_{k=0}^{n} c_k(z^k - z^{-1-k})$ *and* $sgn(A_N) = -1$.

$c_0, \ldots, c_n$ *are uniquely determined by the following recursive formula:* $c_n := \sqrt{|A_N|}$ *and*

$$c_{n-j} := \frac{1}{2c_n}\left[sgn(A_N)A_{N-j} - \sum_{k=n-j+1}^{n-1} c_k c_{2n-j-k}\right], \qquad j = 1, 2, \ldots, n. \tag{2.2}$$

Moreover, if $A(z) = d(z)d(1/z)$ *for an (anti)symmetric Laurent polynomial* d *with real coefficients, then we must have* $d(z) = \pm z^k d_A(z)$ *for some* $k \in \mathbb{Z}$. *Therefore, the symmetry type of the filter* d *is completely determined by the degree of* A *and the sign of the leading term of* A.

Proof. If a Laurent polynomial d is (anti)symmetric and satisfies $A(z) = d(z)d(1/z)$, then it is easy to see that $d(z) = \pm z^k d_A(z)$ for some $k \in \mathbb{Z}$. By comparing the coefficients of $A(z)$ and $d_A(z)d_A(1/z)$, all the claims can be easily verified. □

A similar algorithm for Proposition 2.2 also appeared in [16].

For a matrix M, we denote by $M_{j,k}$ the (j,k)-entry of the matrix M. For a Laurent polynomial p, we denote by $Z(p, z_0)$ the multiplicity of zeros of p at $z = z_0$, that is,

$$Z(p, z_0) = \sup\{n \in \mathbb{N} \cup \{0\} \ : \ (z - z_0)^n \,|\, p(z)\}. \tag{2.3}$$

Now we are ready to state the main results of this paper.

THEOREM 2.3. *Let* A, B, *and* C *be (anti)symmetric Laurent polynomials with real coefficients. Denote a* 2×2 *matrix* M *by*

$$M(z) = \begin{bmatrix} A(z) & B(z) \\ B(1/z) & C(z) \end{bmatrix}, \qquad z \in \mathbb{C}\backslash\{0\}. \tag{2.4}$$

Then there exist (anti)symmetric Laurent polynomials u_1, u_2, v_1, v_2 with real coefficients such that

$$U(z)U(1/z)^T = M(z) \qquad \forall\, z \in \mathbb{C}\backslash\{0\} \quad \textit{with} \qquad U(z) := \begin{bmatrix} u_1(z) & v_1(z) \\ u_2(z) & v_2(z) \end{bmatrix} \tag{2.5}$$

and

$$[Su_1](z)[Sv_2](z) = [Sv_1](z)[Su_2](z), \qquad z \in \mathbb{C}\backslash\{0\}, \tag{2.6}$$

if and only if all the following conditions are satisfied:

(a) *The matrix $M(z)$ is positive semidefinite (that is, $M(z) \geqslant 0$) for all $z \in \mathbb{T}$.*

(b) $\det M(z) = d(z)d(1/z)$ *for some (anti)symmetric Laurent polynomial d with real coefficients.*

(c) *Define $g = gcd(A, B, C)$. If both B and d are not identically zero, then the matrix M satisfies the following "gcd" condition; that is, one of the following conditions must be true:*

1. *If $[SB](z)[Sd](z) = z^{2n}$ for some $n \in \mathbb{Z}$, then $Z(g, x)$ is an even number for every $x \in (-1, 0) \cup (0, 1)$.*
2. *If $[SB](z)[Sd](z) = z^{2n+1}$ for some $n \in \mathbb{Z}$, then $Z(g, x)$ is an even number for every $x \in (0, 1)$.*
3. *If $[SB](z)[Sd](z) = -z^{2n}$ for some $n \in \mathbb{Z}$, then there is no condition on g.*
4. *If $[SB](z)[Sd](z) = -z^{2n+1}$ for some $n \in \mathbb{Z}$, then $Z(g, x)$ is an even number for every $x \in (-1, 0)$.*

We shall prove Theorem 2.3 in section 5 in a constructive way and a step-by-step algorithm (see Algorithm 5.1) will be given to construct the desired filters u_1, u_2, v_1, v_2 from the matrix M. We shall also show that the "gcd" condition in Theorem 2.3 cannot be removed. Note that by Proposition 2.1 and (2.5), it is easy to see that when $B \not\equiv 0$, (2.6) is equivalent to

$$\frac{[Su_1](z)}{[Su_2](z)} = [SB](z) = \frac{[Sv_1](z)}{[Sv_2](z)}, \qquad z \in \mathbb{C}\backslash\{0\}. \tag{2.7}$$

As an application of Theorem 2.3 to symmetric framelet filter banks, we have the following result for constructing symmetric MRA tight wavelet frames with two generators.

THEOREM 2.4. *Let $\phi \in L_2(\mathbb{R})$ be a refinable function satisfying $\hat{\phi}(2\xi) = a(e^{-i\xi})\hat{\phi}(\xi)$ for a symmetric Laurent polynomial a with real coefficients such that $a(1) = 1$. Let Θ be a Laurent polynomial with real coefficients such that $\Theta(z) = \Theta(1/z)$ and $\Theta(1) = 1$. Let M_Θ be defined in* (1.6)*. Then there exist two (anti)symmetric Laurent polynomials a^1 and a^2 with real coefficients such that* (1.5) *in Theorem* 1.1 *holds with $r = 2$ if and only if the following conditions are satisfied:*

(a) *$M_\Theta(z) \geqslant 0$ for all $z \in \mathbb{T}$. (This condition can be replaced by $\Theta(z) \geqslant 0$ for all $z \in \mathbb{T}$.)*

(b) $\det M_\Theta(z) = d(z^2)d(z^{-2})$ *for an (anti)symmetric Laurent polynomial d with real coefficients.*

(c) *Define $g(z^2) = gcd([M_\Theta]_{1,1}, [M_\Theta]_{1,2}, [M_\Theta]_{2,2})$. The matrix M_Θ satisfies the following "gcd" condition; that is, one of the following conditions must be true:*

1. *If $[Sa](-z)[Sd](z) = z^{2n+1}$ for some $n \in \mathbb{Z}$, then $Z(g, x)$ is an even number for every $x \in (-1, 0) \cup (0, 1)$.*
2. *If $[Sa](-z)[Sd](z) = z^{2n}$ for some $n \in \mathbb{Z}$, then $Z(g, x)$ is an even number for every $x \in (0, 1)$.*

3. *If* $[Sa](-z)[Sd](z) = -z^{2n+1}$ *for some* $n \in \mathbb{Z}$*, then there is no condition on* g*.*
4. *If* $[Sa](-z)[Sd](z) = -z^{2n}$ *for some* $n \in \mathbb{Z}$*, then* $Z(g,x)$ *is an even number for every* $x \in (-1,0)$*.*

Proof. Let us make some connections to Theorem 2.3 first. With $r = 2$, (1.5) becomes

$$W(z)W(1/z)^T = M_\Theta(z), \quad \text{where} \quad W(z) = \begin{bmatrix} a^1(z) & a^2(z) \\ a^1(-z) & a^2(-z) \end{bmatrix}. \tag{2.8}$$

Since the mask a is symmetric, we have $[Sa](z) = z^k$ for some $k \in \mathbb{Z}$. Inspired by the idea of polyphase decomposition, we define

$$P(z) := \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & 1 \\ z & -z \end{bmatrix} \text{ if } k \text{ is even}; \quad P(z) := \frac{1}{2}\begin{bmatrix} 1+z & 1-z \\ 1-z & 1+z \end{bmatrix} \text{ if } k \text{ is odd}. \tag{2.9}$$

Then $P(z)P(1/z)^T = I_2$ and $P(-z) = P(z)J_2$, where

$$J_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Now (2.8) can be rewritten as

$$U(z)U(1/z)^T = M(z) \tag{2.10}$$

with

$$U(z^2) = \tilde{W}(z) := P(z)W(z), \quad M(z^2) = \tilde{M}(z) := P(z)M_\Theta(z)P(1/z)^T. \tag{2.11}$$

When k is even, by computation we have

$$\tilde{W}(z) = \frac{\sqrt{2}}{2}\begin{bmatrix} a^1(z)+a^1(-z) & a^2(z)+a^2(-z) \\ za^1(z)-za^1(-z) & za^2(z)-za^2(-z) \end{bmatrix} \tag{2.12}$$

and

$$[\tilde{M}(z)]_{1,2} = \frac{1}{2z}\big(\Theta(z) - \Theta(-z) - \Theta(z^2)[a(z)+a(-z)][a(1/z)-a(-1/z)]\big).$$

It is easy to see that $\tilde{W}(-z) = \tilde{W}(z)$ and

$$\tilde{M}(-z) = P(z)J_2M_\Theta(-z)J_2P(1/z)^T = P(z)M_\Theta(z)P(1/z)^T = \tilde{M}(z).$$

So, U and M are well defined. Moreover, it is easy to see that $M_{1,2} \not\equiv 0$ and $[SM_{1,2}](z) = z^{-1}$.

When k is odd, by computation we have

$$\tilde{W}(z) = \frac{1}{2}\begin{bmatrix} (1+z)a^1(z)+(1-z)a^1(-z) & (1+z)a^2(z)+(1-z)a^2(-z) \\ (1-z)a^1(z)+(1+z)a^1(-z) & (1-z)a^2(z)+(1+z)a^2(-z) \end{bmatrix} \tag{2.13}$$

and

$$\begin{aligned}\tilde{M}_{1,2}(z) = \frac{1}{4}(z-1/z)[\Theta(z)-\Theta(-z)] - \frac{1}{4}\Theta(z^2)[(1+z)a(z)+(1-z)a(-z)] \\ \times [(1-1/z)a(1/z)+(1+1/z)a(-1/z)].\end{aligned}$$

It is clear that $\tilde{W}(-z) = \tilde{W}(z)$ and $\tilde{M}(-z) = \tilde{M}(z)$. So, U and M are well defined. Moreover, it is easy to see that $M_{1,2} \not\equiv 0$ and $[SM_{1,2}](z) = -1$.

By the definition of P and the definition (2.11), we have

$$\det M(z^2) = \det M_\Theta(z)$$

and

$$g(z) = \gcd(M_{1,1}, M_{1,2}, M_{2,2}),$$

where

$$g(z^2) = \gcd([M_\Theta]_{1,1}, [M_\Theta]_{1,2}, [M_\Theta]_{2,2}).$$

By the discussion above, we can clearly see the relation between conditions (a), (b), and (c) in Theorem 2.3 and conditions (a), (b), and (c) in this theorem, respectively. Based on the relation, we will prove the necessity and sufficiency, respectively.

Necessity. Suppose that there exist two (anti)symmetric Laurent polynomials a^1 and a^2 with real coefficients such that (1.5) holds; by (2.8) we have $M_\Theta(z) \geqslant 0$ for all $z \in \mathbb{T}$ and therefore condition (a) holds. Note that $\det W(-z) = -\det W(z)$. Thus, we can define a Laurent polynomial d by $d(z^2) = z \det W(z)$. Clearly,

$$\det M_\Theta(z) = \det W(z) \det W(1/z) = d(z^2) d(z^{-2}).$$

We now show that d is (anti)symmetric. Since a^1 and a^2 are (anti)symmetric, we have $[Sa^1](z) = \varepsilon_1 z^{k_1}$ and $[Sa^2](z) = \varepsilon_2 z^{k_2}$ for some $\varepsilon_1, \varepsilon_2 \in \{-1, 1\}$ and $k_1, k_2 \in \mathbb{Z}$. By (2.8) and (1.6), we have

$$a^1(z)a^1(-1/z) + a^2(z)a^2(-1/z) = [M_\Theta]_{1,2}(z) = -\Theta(z^2)a(z)a(-1/z). \tag{2.14}$$

Note that

$$S[a^1(z)a^1(-1/z)] = \varepsilon_1 z^{k_1} \varepsilon_1 (-1/z)^{k_1} = (-1)^{k_1}$$

and similarly, $S[a^2(z)a^2(-1/z)] = (-1)^{k_2}$. Since

$$S[\Theta(z^2)a(z)a(-1/z)] = S[\Theta(z^2)] S[a(z)a(-1/z)] = (-1)^k,$$

by a simple argument, it follows from (2.14) that

$$(-1)^{k_1} = (-1)^{k_2} = (-1)^k. \tag{2.15}$$

(Note that there are at least two even (or odd) numbers among k_1, k_2, and k. Say, k_1 and k_2 are even. Then by item (4) in Proposition 2.1, we conclude that $(-1)^{k_1} = (-1)^{k_2} = (-1)^k$.) Note that $\det W(z) = a^1(z)a^2(-z) - a^1(-z)a^2(z)$. Since

$$\begin{aligned} S[a^1(z)a^2(-z)] &= \varepsilon_1 \varepsilon_2 (-1)^{k_2} z^{k_1+k_2} = \varepsilon_1 \varepsilon_2 (-1)^k z^{k_1+k_2} \\ &= \varepsilon_1 \varepsilon_2 (-1)^{k_1} z^{k_1+k_2} = S[a^1(-z)a^2(z)], \end{aligned}$$

by Proposition 2.1, we conclude that

$$S[\det W(z)] = \varepsilon_1 \varepsilon_2 (-1)^k z^{k_1+k_2}. \tag{2.16}$$

So, det$W$ is (anti)symmetric and therefore, by $d(z^2) = z\text{det}W(z)$, d is (anti)symmetric. Hence condition (b) holds.

Recall that $[Sa](z) = z^k$. When k is even, by Proposition 2.1 and the fact that $(-1)^{k_1} = (-1)^{k_2} = (-1)^k = 1$, it follows from (2.12) that

$$[S\tilde{W}_{1,1}](z) = \varepsilon_1 z^{k_1}, \quad [S\tilde{W}_{1,2}](z) = \varepsilon_2 z^{k_2}, \quad [S\tilde{W}_{2,1}](z) = \varepsilon_1 z^{2+k_1},$$
$$[S\tilde{W}_{2,2}](z) = \varepsilon_2 z^{2+k_2}, \quad \text{and} \quad [S\tilde{M}_{1,2}](z) = z^{-2}.$$

Thus, when k is even, (2.5) and (2.6) are satisfied. Since $P(z)P(1/z)^T = I_2$, we must have $g = \gcd(M_{1,1}, M_{1,2}, M_{2,2})$. Note that

$$[SM_{1,2}](z)[Sd](z) = z^{-1}[Sd](z) = z^{-1-k}[Sa](-z)[Sd](z)$$

and k is an even integer. Therefore, by Theorem 2.3, condition (c) must be true.

When k is odd, by Proposition 2.1 and the fact that $(-1)^{k_1} = (-1)^{k_2} = (-1)^k = -1$, it follows from (2.13) that

$$[S\tilde{W}_{1,1}](z) = \varepsilon_1 z^{1+k_1}, \quad [S\tilde{W}_{1,2}](z) = \varepsilon_2 z^{1+k_2}, \quad [S\tilde{W}_{2,1}](z) = -\varepsilon_1 z^{1+k_1},$$
$$[S\tilde{W}_{2,2}](z) = -\varepsilon_2 z^{1+k_2}, \quad \text{and} \quad [S\tilde{M}_{1,2}](z) = -1.$$

Thus, when k is odd, (2.5) and (2.6) are satisfied. Note that

$$[SM_{1,2}](z)[Sd](z) = -[Sd](z) = -(-z)^{-k}[Sa](-z)[Sd](z) = z^{-k}[Sa](-z)[Sd](z)$$

and k is an odd integer. Therefore, by Theorem 2.3, condition (c) must be true.

Sufficiency. Suppose that conditions (a), (b), and (c) in this theorem are satisfied. From the discussion before the necessity part, applying Theorem 2.3 on $M(z)$, we know that there exist (anti)symmetric Laurent polynomials u_1, u_2, v_1, v_2 with real coefficients such that (2.5) and (2.7) hold. Define

$$\begin{bmatrix} a^1(z) & a^2(z) \\ a^1(-z) & a^2(-z) \end{bmatrix} := P(1/z)^T U(z^2) = P(1/z)^T \begin{bmatrix} u_1(z^2) & v_1(z^2) \\ u_2(z^2) & v_2(z^2) \end{bmatrix}.$$

We show that a^1 and a^2 must be (anti)symmetric. Since $[Sa](z) = z^k$, when k is even, we have

$$a^1(z) = \frac{\sqrt{2}}{2}[u_1(z^2) + u_2(z^2)/z] \quad \text{and} \quad a^2(z) = \frac{\sqrt{2}}{2}[v_1(z^2) + v_2(z^2)/z].$$

By $[SM_{1,2}](z) = z^{-1}$, it follows from (2.7) that

$$S(u_1(z^2)) = S(M_{1,2}(z^2))S(u_2(z^2)) = z^{-2}S(u_2(z^2)) = S(u_2(z^2)/z)$$

and $S(v_1(z^2)) = S(v_2(z^2)/z)$. By Proposition 2.1, we have $[Sa^1](z) = [Su_1](z^2)$ and $[Sa^2](z) = [Sv_1](z^2)$. Since u_1 and v_1 are (anti)symmetric, so are the Laurent polynomials a^1 and a^2.

When k is odd, we have

$$a^1(z) = [(1 + 1/z)u_1(z^2) + (1 - 1/z)u_2(z^2)]/2$$

and

$$a^2(z) = [(1 + 1/z)v_1(z^2) + (1 - 1/z)v_2(z^2)]/2.$$

By $[SM_{1,2}](z) = -1$, it follows from (2.7) that

$$\begin{aligned} S((1+1/z)u_1(z^2)) &= z^{-1}S(u_1(z^2)) \\ &= z^{-1}S(M_{1,2}(z^2))S(u_2(z^2)) \\ &= S((1-1/z)u_2(z^2)) \end{aligned}$$

and $S((1+1/z)v_1(z^2)) = S((1-1/z)v_2(z^2))$. By Proposition 2.1, we deduce that $[Sa^1](z) = z^{-1}[Su_1](z^2)$ and $[Sa^2](z) = z^{-1}[Sv_1](z^2)$. Since both u_1 and v_1 are (anti)symmetric, so are the Laurent polynomials a^1 and a^2. Now it is straightforward to verify that (2.8) holds. □

In order to construct symmetric framelet filter banks with two high-pass filters, by the proof of Theorem 2.4, we present the following algorithm.

ALGORITHM 2.5. *Let a be a symmetric Laurent polynomial with real coefficients such that $a(1) = 1$ (that is, a is a low-pass filter). Suppose that we have a Laurent polynomial Θ such that all the conditions in Theorem* 2.4 *are satisfied.*

(1) *Compute the symmetry center of the low-pass filter a: $[Sa](z) := a(z)/a(1/z) = z^k$ for some integer k. Define the 2×2 matrix P in* (2.9) *according to the parity of the integer k.*

(2) *Calculate the 2×2 matrix $M(z^2) := P(z)M_\Theta(z)P(1/z)^T$, where M_Θ is defined in* (1.6).

(3) *Using Algorithm* 5.1 *in section* 5 *to split the matrix M into the desired form:*

$$M(z) = \begin{bmatrix} u_1(z) & v_1(z) \\ u_2(z) & v_2(z) \end{bmatrix} \begin{bmatrix} u_1(1/z) & u_2(1/z) \\ v_1(1/z) & v_2(1/z) \end{bmatrix}$$

and

$$[Su_1](z)[Sv_2](z) = [Su_2](z)[Sv_1](z).$$

In most cases $g(z^2) = gcd([M_\Theta]_{1,1}, [M_\Theta]_{1,2}, [M_\Theta]_{2,2}) = 1$ and consequently, by solving a system of linear equations, we have all the symmetric filters u_1, u_2, v_1, v_2 by Algorithm 5.1.

(4) *Obtain the symmetric high-pass filters a^1 and a^2 by*

$$\begin{aligned} a^1(z) &:= P_{1,1}(1/z)u_1(z^2) + P_{2,1}(1/z)u_2(z^2), \\ a^2(z) &:= P_{1,1}(1/z)v_1(z^2) + P_{2,1}(1/z)v_2(z^2). \end{aligned}$$

Then (2.8) *holds and we have a symmetric framelet filter bank consisting of a low-pass filter a and two high-pass filters a^1 and a^2.*

In order to design a desired filter Θ such that all the conditions in Theorem 2.4 are satisfied, quite often one constructs a Θ such that $\Theta(1) = 1$, $\Theta(z) \geqslant 0$ for all $z \in \mathbb{T}$ and

$$\begin{aligned} \det M(z^2) &= \Theta(z)\Theta(-z) - \Theta(z^2)[\Theta(z)a(-z)a(-1/z) + \Theta(-z)a(z)a(1/z)] \\ &= d(z^2)d(z^{-2}), \end{aligned}$$

where d is determined in Proposition 2.2. In most cases the "gcd" condition in Theorem 2.4 is automatically satisfied. More explicitly, we usually set $\Theta(z) = 1+c_1w+\cdots+c_nw^n, w = (2-z-1/z)/4$ with some unknown parameters $c_1, \ldots, c_n$. So, we automatically have $\Theta(1) = 1$. Then we obtain some equations for the unknowns $c_1, \ldots, c_n$ from

the condition $\det M(z^2) = d(z^2)d(z^{-2})$ by Proposition 2.2. Solving such equations for the unknowns $c_1, \ldots, c_n$, we see that the desired condition $\det M(z^2) = d(z^2)d(z^{-2})$ holds. Finally, we check the two conditions $\Theta(z) \geqslant 0$ and the "gcd" condition which quite often turn out to be satisfied automatically.

Suppose that the low-pass filter a is given. Theorem 2.4 gives us a necessary and sufficient condition on Θ to construct a symmetric framelet with two high-pass FIR filters. If we have found a desired Θ such that conditions (a), (b), and (c) in Theorem 2.4 hold, then we can use Algorithm 2.5 to construct two symmetric high-pass filters a^1 and a^2. Since we have some freedom in constructing a^1 and a^2 from a and Θ, it is of interest to know what are all the possible symmetry types for these two high-pass filters a^1 and a^2. We shall see in the following result that the symmetry types of the high-pass filters a^1 and a^2 are completely determined by a and Θ.

THEOREM 2.6. *Let $\phi \in L_2(\mathbb{R})$ be a refinable function satisfying $\hat{\phi}(2\xi) = a(e^{-i\xi})\hat{\phi}(\xi)$ for a symmetric Laurent polynomial a with real coefficients such that $a(1) = 1$. Let Θ be a Laurent polynomial with real coefficients such that $\Theta(z) = \Theta(1/z)$ and $\Theta(1) = 1$. Suppose that conditions* (a), (b), *and* (c) *in Theorem* 2.4 *are satisfied. Let a^1 and a^2 be two (anti)symmetric Laurent polynomials with real coefficients such that* (1.5) *in Theorem* 1.1 *are satisfied with $r = 2$. Denote*

$$[Sa](z) = z^k, \qquad [Sa^1](z) = \varepsilon_1 z^{k_1}, \qquad [Sa^2](z) = \varepsilon_2 z^{k_2}$$

for some $\varepsilon_1, \varepsilon_2 \in \{-1, 1\}$ and for some integers k, k_1, and k_2. Then k, k_1, and k_2 have the same parity (that is, $k_1 - k$ and $k_2 - k$ are even integers) and one of the following two cases must be true:

(a) *If $[Sd](1) = (-1)^{k+1}$, then $\varepsilon_1\varepsilon_2 = -1$; that is, either $\varepsilon_1 = -1$, $\varepsilon_2 = 1$ (ψ^1 is antisymmetric and ψ^2 is symmetric) or $\varepsilon_1 = 1$, $\varepsilon_2 = -1$ (ψ^1 is antisymmetric and ψ^2 is symmetric);*

(b) *If $[Sd](1) = (-1)^k$, then $\varepsilon_1 = \varepsilon_2 = (-1)^n$, where $n = Z(h, 1)/2$ and $h(z) := \Theta(z) - \Theta(z^2)a(z)a(1/z)$.*

In conclusion, up to a trivial switch of the two high-pass filters a^1 and a^2, the symmetry types of the filters a^1 and a^2 are completely determined by the low-pass filter a and the filter Θ.

Proof. We use the proof of the necessity part of Theorem 2.4 . As in the proof of Theorem 2.4, we must have

$$(-1)^{k_1} = (-1)^{k_2} = (-1)^k.$$

Therefore, both a^1 and a^2 have even degrees if a has an even degree, or, both a^1 and a^2 have odd degrees if a has an odd degree. Thus, we only need to prove that up to a trivial switch of the two high-pass filters a^1 and a^2, the numbers ε_1 and ε_2 are completely determined by the low-pass filter a and the filter Θ. In the proof of the necessity part of Theorem 2.4, we proved that (2.16) must be true. By Proposition 2.2 and $\det M_\Theta(z) = d(z^2)d(z^{-2})$, we know that $[Sd](1)$ depends only on a and Θ. Consequently, we can assume that $d(z^2) = z\det W(z)$. Hence, it follows from (2.16) that we must have

$$\varepsilon_1\varepsilon_2 = [Sd](1)[Sa](-1). \tag{2.17}$$

If $[Sd](1)[Sa](-1) = -1$, then it follows from (2.17) that $\varepsilon_1\varepsilon_2 = -1$; therefore, we have either $\varepsilon_1 = -1$, $\varepsilon_2 = 1$ or $\varepsilon_1 = 1$, $\varepsilon_2 = -1$. If $[Sd](1)[Sa](-1) = 1$, it follows from (2.17) that $\varepsilon_1\varepsilon_2 = 1$. To complete the proof, it suffices to consider the case

$[Sd](1)[Sa](-1) = 1$. In this case, we must have $\varepsilon_1 = \varepsilon_2$ since $\varepsilon_1\varepsilon_2 = 1$ and $\varepsilon_1, \varepsilon_2 \in \{-1, 1\}$. It is easy to see that

$$\varepsilon_1 = (-1)^{Z(a^1,\,1)} \quad \text{and} \quad \varepsilon_2 = (-1)^{Z(a^2,\,1)}.$$

Define

$$n := \min\big(Z(a^1, 1), Z(a^2, 1)\big).$$

Then we have $\varepsilon_1 = \varepsilon_2 = (-1)^n$ and

$$\lim_{z\to 1} \frac{a^1(z)a^1(1/z) + a^2(z)a^2(1/z)}{|z-1|^{2n}} \in (0, \infty).$$

Define $h(z) := \Theta(z) - \Theta(z^2)a(z)a(1/z)$. By (2.8) and (1.6), we have

$$a^1(z)a^1(1/z) + a^2(z)a^2(1/z) = h(z).$$

Thus, we have $2n = Z(h, 1)$, that is, $n = Z(h, 1)/2$. Hence, n depends only on a and Θ. Therefore, ε_1 and ε_2 depend only on a and Θ. ∎

3. Some examples of symmetric framelet filter banks. First, we illustrate that the "gcd" condition in Theorem 2.4 cannot be removed. Then by Algorithm 2.5 we provide several examples of symmetric framelet filter banks with two high-pass filters. In Theorem 2.4, the "gcd" condition seems unnatural. One may conjecture that the "gcd" condition will be automatically satisfied if $M_\Theta(z) \geqslant 0$ for all $z \in \mathbb{T}$ and $\det M_\Theta(z) = d(z^2)d(z^{-2})$ holds for some (anti)symmetric Laurent polynomial d. The following example shows that this conjecture is not true.

Example 3.1. Let the low-pass filter a be given by

$$a(z) := \frac{1}{4}(1+z)^2[1 + c_1(2 - z - z^{-1})/2],$$

where $c_1 \approx 0.07391$ is a root of $x^8+8x^7+35x^6+58x^5-10x^4-72x^3-x^2+14x-1 = 0$. By a simple calculation, it is easy to verify that the refinable function ϕ with the mask a lies in $L_2(\mathbb{R})$ and in fact is a continuous function. Define $b := c_1^2 + 2c_1 - 1$, $f(z) := 1+b(2-z-z^{-1})/4$ and $\Theta(z) := f(z^2)f(z)$. It is easy to verify that $M_\Theta(z) \geqslant 0$ for all $z \in \mathbb{T}$ and $\det M_\Theta(z) = d(z^2)d(z^{-2})$ for some antisymmetric Laurent polynomial d such that $[Sd](z) = -z^2$. Let $x_0 = 1 + 2(1 - \sqrt{b+1})/b \approx -0.43729 \in (-1, 0)$ which satisfies $f(x_0) = 0$. By a simple computation, we have $Z(g, x_0) = 1$, where $g(z^2) = f(z^2) = \gcd([M_\Theta]_{1,1}, [M_\Theta]_{1,2}, [M_\Theta]_{2,2})$. Since $[Sa](-z)[Sd](z) = z^2(-z^2) = -z^4$, the "gcd" condition fails while conditions (a) and (b) in Theorem 2.4 are satisfied. Therefore, the "gcd" condition in Theorem 2.4 cannot be removed.

In the following, let us apply Algorithm 2.5 to obtain several examples of symmetric framelet filter banks with two high-pass filters.

Example 3.2. Let $\phi = B_3$ be the B-Spline function of order 3. It is known that the low-pass filter for B_3 is $a(z) = (z+1)^3/8$. Define

$$\Theta(z) := 1 + w + 13/15\, w^2 + c_1 w^3 + c_2 w^4 \quad \text{with} \quad w = (2 - z - z^{-1})/4.$$

In order to satisfy the condition $\det M_\Theta(z) = d(z^2)d(z^{-2})$ for some (anti)symmetric Laurent polynomial d, we find that c_2 must be one of the 6 real roots of a polynomial of degree 16 and c_1 can be expressed as a rational polynomial with variable c_2. For simplicity, we present them in decimal notation: $c_1 \approx -0.9515104959378669$ and $c_2 \approx 3.803127158568155$. It is easy to check that $g = 1$ and all the conditions in Theorem 2.4 are satisfied. By Algorithms 2.5 and 5.1, solving a system of linear

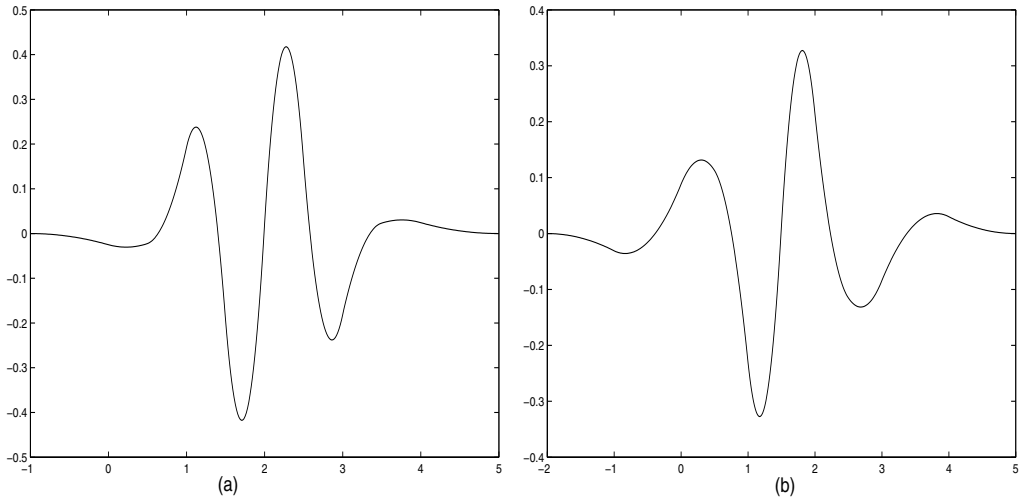


FIG. 1. (a) *is the graph of* ψ^1. (b) *is the graph of* ψ^2. $\{\psi^1, \psi^2\}$ *in Example* 3.2 *generates a symmetric tight wavelet frame with* 3 *vanishing moments.*

equations, we have the high-pass filters a^1 and a^2 as follows:

$$\begin{aligned} a^1(z) := z(z-1)^3\Big[&0.012317964188125 51(z^3+z^{-3}) + 0.07390778512875306(z^2+z^{-2}) \\ &+ 0.1935907748598208(z+z^{-1}) - 0.01145080836662162\Big], \\ a^2(z) := (z-1)^3\Big[&0.01523563127546168(z^4+z^{-4}) + 0.09141378765277004(z^3+z^{-3}) \\ &+ 0.2159429726473255(z^2+z^{-2}) + 0.2291636466016358(z+z^{-1}) \\ &+ 0.06272019447988098\Big]. \end{aligned}$$

Therefore, $\{\psi^1, \psi^2\}$, which is defined in Theorem 1.1, generates a symmetric tight wavelet frame and has 3 vanishing moments. See Figure 1 for their graphs.

Example 3.3. Let $\phi = B_4$ be the B-Spline function of order 4. The low-pass filter for B_4 is $a(z) = (z+1)^4/16$. Define

$$\Theta(z) := 1 + 3/4\, w + 62/45\, w^2 + c_1 w^3 + c_2 w^4 + c_3 w^5 \quad \text{with} \quad w = (2 - z - z^{-1})/4.$$

In order to satisfy the condition $\det M_\Theta(z) = d(z^2)d(z^{-2})$ for some (anti)symmetric Laurent polynomial d, we find a solution $\{c_1, c_2, c_3\}$ in decimal notation as follows:

$$\begin{aligned} &c_1 \approx -0.8755856554740179, \qquad c_2 \approx -0.09842565346701244, \\ &c_3 \approx 0.0009697256495300811. \end{aligned}$$

Then $g = 1$ and all the conditions in Theorem 2.4 hold. By Algorithms 2.5 and 5.1, solving a system of linear equations, we have the high-pass filters a^1 and a^2 as

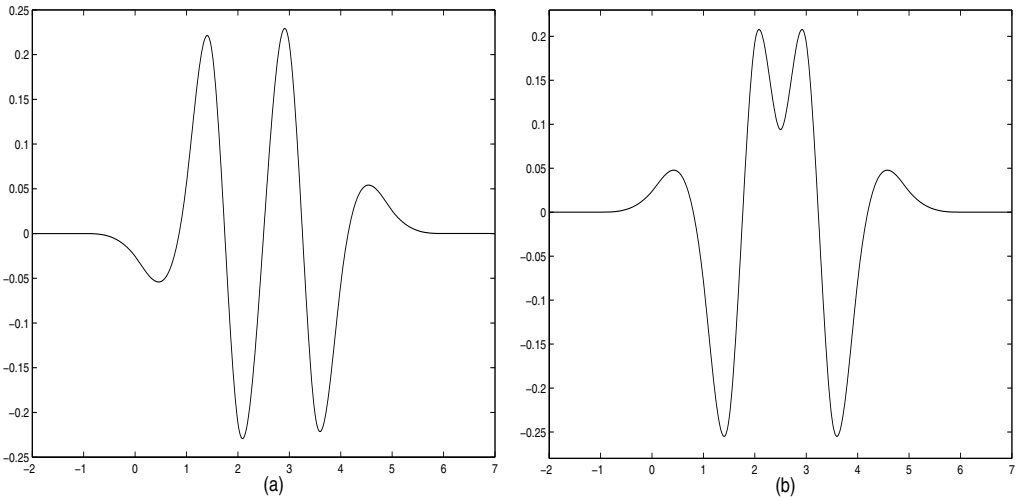


FIG. 2. (a) *is the graph of* ψ^1. (b) *is the graph of* ψ^2. $\{\psi^1, \psi^2\}$ *in Example* 3.3 *generates a symmetric tight wavelet frame with* 3 *vanishing moments.*

follows:

$$\begin{aligned} a^1(z) :=& z(z+1)(z-1)^3\Big[0.00002100045515458106(z^5+z^{-5}) \\ &+ 0.0001260027309274863(z^4+z^{-4}) + 0.01944570184560223(z^3+z^{-3}) \\ &+ 0.1152041792127928(z^2+z^{-2}) + 0.2275150394894326(z+z^{-1}) \\ &+ 0.009838194257376166\Big], \\ a^2(z) :=& z(z-1)^4\Big[0.00006434461049978000(z^5+z^{-5}) \\ &+ 0.0005147568839982400(z^4+z^{-4}) + 0.01966520045452812(z^3+z^{-3}) \\ &+ 0.1465117090722619(z^2+z^{-2}) + 0.4466955026709126(z+z^{-1}) \\ &+ 0.5353777065261440\Big]. \end{aligned}$$

Therefore, $\{\psi^1, \psi^2\}$, which is defined in Theorem 1.1, generates a symmetric tight wavelet frame and has 3 vanishing moments. See Figure 2 for their graphs.

Example 3.4. The low-pass filter a is given by

$$a(z) = z^{-2}(z+1)^4(4-z-z^{-1})/32 = -(z^3+z^{-3})/32 + 9(z+z^{-1})/32 + 1/2.$$

Define

$$\Theta(z) := 1 + 2/5\, w^2 + 44/315\, w^3 + c_1 w^4 + c_2 w^5 + c_3 w^6 + c_4 w^7 + c_5 w^8 + c_6 w^9$$

with $w := (2-z-z^{-1})/4$. In order to satisfy $\det M_\Theta(z) = d(z^2)d(z^{-2})$ for some (anti)symmetric Laurent polynomial d, we find a solution $\{c_1, c_2, c_3, c_4, c_5, c_6\}$ in decimal notation as follows:

$$\begin{aligned} &c_1 \approx -0.5391476369353669, && c_2 \approx 0.03123065991448046, \\ &c_3 \approx 0.1404437899699654, && c_4 \approx -0.008183355709257437, \\ &c_5 \approx -0.02305770106687993, && c_6 \approx 0.005166592059270131. \end{aligned}$$

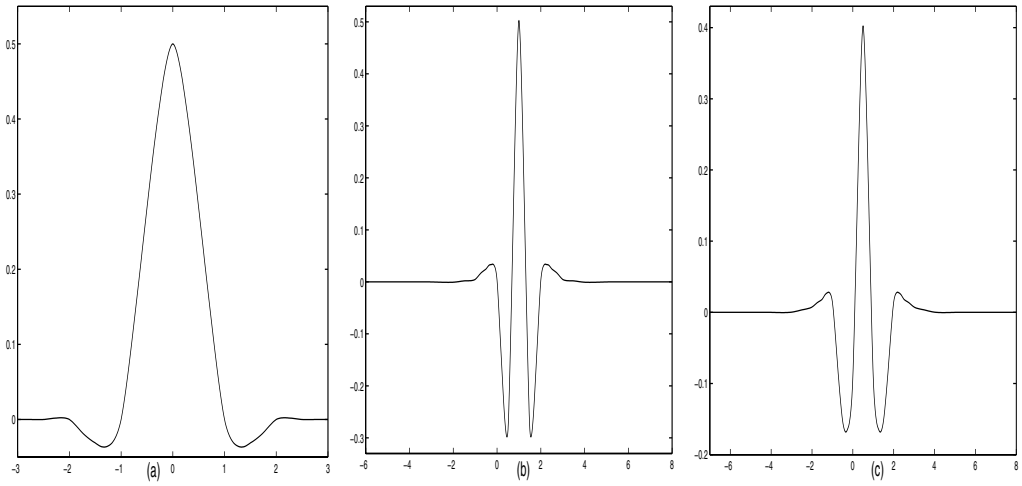


FIG. 3. (a) *is the graph of the interpolating refinable function* ϕ. (b) *is the graph of* ψ^1. (c) *is the graph of* ψ^2. $\{\psi^1, \psi^2\}$ *in Example* 3.4 *generates a symmetric tight wavelet frame with* 4 *vanishing moments.*

It is easy to check that $g = 1$ and all the conditions in Theorem 2.4 hold. By Algorithms 2.5 and 5.1, solving a system of linear equations, we have the high-pass filters a^1 and a^2 as follows:

$$\begin{aligned}
a^1(z) :=& (z-1)^4\Big[0.000009949295438893275(z^9+z^{-9})\\
&+0.000039797181755573 10(z^8+z^{-8})+0.000053494253603311 52(z^7+z^{-7})\\
&-0.000144197621386911 8(z^6+z^{-6})-0.001526840787475249(z^5+z^{-5})\\
&-0.005764716919552544(z^4+z^{-4})-0.012645206603521 71(z^3+z^{-3})\\
&-0.017243945163087 53(z^2+z^{-2})+0.010390964099455 11(z+z^{-1})\\
&+0.1033772717787854\Big],\\
a^2(z) :=& (z-1)^4\Big[0.000004387146598246904(z^9+z^{-11})\\
&+0.000017548586392987 62(z^8+z^{-10})-0.000007220506808539094(z^7+z^{-9})\\
&-0.000186819304771044 9(z^6+z^{-8})-0.001033777502667078(z^5+z^{-7})\\
&-0.002874902341160608(z^4+z^{-6})-0.002673048014126028(z^3+z^{-5})\\
&+0.009978772639517269(z^2+z^{-4})+0.063882503735930 19(z+z^{-3})\\
&+0.2021738981781012(1+z^{-2})+0.3153550969816685z^{-1}\Big].
\end{aligned}$$

Therefore, $\{\psi^1, \psi^2\}$, which is defined in Theorem 1.1, generates a symmetric tight wavelet frame and has 4 vanishing moments. See Figure 3 for their graphs.

4. Some auxiliary results. In order to prove Theorem 2.3, in this section we establish some auxiliary results. The following result generalizes [2, Theorem 4] by taking into account symmetry.

THEOREM 4.1. *Let* A, B, *and* C *be (anti)symmetric Laurent polynomials with real coefficients. Let* M *be defined in* (2.4). *Suppose that* $A(z) = A_0 + \sum_{k=1}^{N} A_k(z^k + z^{-k})$ *with* $A_N \neq 0$, $M(z) \geqslant 0$ *for all* $z \in \mathbb{T}$ *and* $\det M(z) = d(z)d(1/z)$ *for some*

(anti)symmetric Laurent polynomial d with real coefficients. If A and B have no common zeros in $\mathbb{C}\backslash\{0\}$, then there exist four (anti)symmetric Laurent polynomials u_1, u_2, v_1, v_2 with real coefficients such that (2.5) *and* (2.7) *are satisfied with the degrees of u_1 and v_1 being at most N. In fact, if u_1, u_2, v_1, v_2 are (anti)symmetric Laurent polynomials with real coefficients such that the degrees of u_1 and v_1 are at most N,* (2.7) *holds, and $\{u_1, u_2, v_1, v_2\}$ is a solution to the system of linear equations*

$$\begin{cases} B(1/z)u_1(z) - d(z)v_1(1/z) - A(z)u_2(z) = 0, \\ B(1/z)v_1(z) + d(z)u_1(1/z) - A(z)v_2(z) = 0, \end{cases} \tag{4.1}$$

with the normalization condition

$$u_1(1)^2 + v_1(1)^2 = A(1), \tag{4.2}$$

then (2.5) *holds.*

Proof. If $B(z) \equiv 0$, by $\gcd(A, B) = 1$, then $A(z)$ must be a positive constant and all the claims can be easily verified by taking $u_1 = \sqrt{A}, u_2 = 0, v_1 = 0$, and $v_2 = d/\sqrt{A}$. So, we can assume that B is not identically zero.

If $d(z) \equiv 0$, then $A(z)C(z) = B(z)B(1/z)$. Since $\gcd(A, B) = 1$ and B is (anti)symmetric, it follows from $A(z)C(z) = B(z)B(1/z) = B(z)^2/[SB](z)$ that A must be a positive constant. All the claims hold by taking $u_1 = \sqrt{A}, u_2 = B(1/z)/\sqrt{A}, v_1 = 0$ and $v_2 = 0$. So, we can assume that d is not identically zero.

In the first part of our proof, let us recall the proof of [2, Theorem 4]. Under the assumption that (2.7) and the degrees of u_1 and v_1 are at most N, we first show that (2.5) is equivalent to the system of linear equations in (4.1) with the condition in (4.2).

Since $M(z) \geqslant 0$ for all $z \in \mathbb{T}$ and $\gcd(A, B) = 1$, if we have $A(z_0) = 0$ for some $z_0 \in \mathbb{T}$, then by condition $M(z_0) \geqslant 0$, we have

$$0 \leqslant \det M(z_0) = A(z_0)C(z_0) - B(z_0)B(1/z_0) = -B(z_0)\overline{B(z_0)} = -|B(z_0)|^2.$$

Hence, $B(z_0) = 0$. Therefore, $(z - z_0) \mid A(z)$ and $(z - z_0) \mid B(z)$. So, $(z - z_0) \mid \gcd(A, B)$ which is a contradiction to the assumption $\gcd(A, B) = 1$. So, $A(z) \neq 0$ for all $z \in \mathbb{T}$. Since $A(z) \geqslant 0$ for all $z \in \mathbb{T}$, we must have $A(z) > 0$ for all $z \in \mathbb{T}$. By Proposition 2.2, without loss of generality, we can assume that $d(z) = \det U(z)$. By $U(z)U(1/z)^T = M(z)$, we have $u_1(1)^2 + v_1(1)^2 = A(1)$ and therefore (4.2) holds. Since $d(z) \not\equiv 0$ and $d(z)U(z)^{-1} = \mathrm{adj}U(z)$, it follows from $U(z)U(1/z)^T = M(z)$ that

$$\begin{aligned} d(z)U(1/z)^T &= d(z)U(z)^{-1}M(z) = [\mathrm{adj}U(z)]M(z) \\ &= \begin{bmatrix} v_2(z) & -v_1(z) \\ -u_2(z) & u_1(z) \end{bmatrix} \begin{bmatrix} A(z) & B(z) \\ B(1/z) & C(z) \end{bmatrix}. \end{aligned}$$

Comparing the $(1, 1)$ and $(2, 1)$-entries of the above matrices, we see that (4.1) holds.

Conversely, let u_1, u_2, v_1, v_2 be (anti)symmetric Laurent polynomials with real coefficients such that (2.7) holds and the degrees of u_1 and v_1 are at most N. If $\{u_1, u_2, v_1, v_2\}$ is a solution to the system of linear equations in (4.1) and satisfies the normalization condition in (4.2), then we show that (2.5) must be true.

Multiplying $u_1(1/z)$ with the first equation and multiplying $v_1(1/z)$ with the second equation in (4.1), by adding them together we have

$$B(1/z)[u_1(z)u_1(1/z) + v_1(z)v_1(1/z)] = A(z)[u_1(1/z)u_2(z) + v_1(1/z)v_2(z)]. \tag{4.3}$$

Since A and B have no common zeros in $\mathbb{C}\backslash\{0\}$, we must have $A(z) \mid [u_1(z)u_1(1/z) + v_1(z)v_1(1/z)]$. That is, there is a Laurent polynomial p such that $u_1(z)u_1(1/z) + v_1(z)v_1(1/z) = p(z)A(z)$. Since the degrees of u_1 and v_1 are at most N and $A(z) = A_0 + \sum_{k=1}^{N} A_k(z^k + z^{-k})$ with $A_N \neq 0$, we conclude that p must be a constant. By (4.2) and $A(1) > 0$, we must further have $p \equiv 1$. Therefore,

$$u_1(z)u_1(1/z) + v_1(z)v_1(1/z) = A(z).$$

It follows from (4.3) that $B(1/z) = u_1(1/z)u_2(z) + v_1(1/z)v_2(z)$ and consequently, $B(z) = u_1(z)u_2(1/z) + v_1(z)v_2(1/z)$. In other words, $[U(z)U(1/z)^T]_{j,k} = [M(z)]_{j,k}$ for all $1 \leqslant j, k \leqslant 2$ except for the case $j = k = 2$.

Multiplying $v_2(z)$ with the first equation and multiplying $u_2(z)$ with the second equation in (4.1), by subtracting the second one from the first one, we have

$$B(1/z)[u_1(z)v_2(z) - u_2(z)v_1(z)] = d(z)[u_1(1/z)u_2(z) + v_1(1/z)v_2(z)] = d(z)B(1/z).$$

So, by $B \not\equiv 0$, $d(z) = u_1(z)v_2(z) - u_2(z)v_1(z) = \det U(z)$. Consequently,

$$\det[U(z)U(1/z)^T] = d(z)d(1/z) = \det M(z).$$

Now it is easy to deduce that $[U(z)U(1/z)^T]_{2,2} = [M(z)]_{2,2}$ from the fact that $\det[U(z)U(1/z)^T] = \det M(z)$ and $[U(z)U(1/z)^T]_{j,k} = [M(z)]_{j,k}$ for all $1 \leqslant j, k \leqslant 2$ except for $j = k = 2$. So (2.5) holds.

In the second part of the proof, let us show the existence of a desirable solution $\{u_1, u_2, v_1, v_2\}$ to the system of linear equations in (4.1) with the normalization condition in (4.2).

First, we demonstrate that there are desirable Laurent polynomials u_1 and v_1 satisfying

$$A(z) \mid [B(1/z)u_1(z) - d(z)v_1(1/z)] \tag{4.4}$$

and

$$[Su_1](z)[Sv_1](z) = [SB](z)[Sd](z). \tag{4.5}$$

Let u_0 and v_0 be two symmetric Laurent polynomials in the following parametric forms:

$$u_0(z) = b_0 + \sum_{j=1}^{h_b} b_j(z^j + z^{-j}) \quad \text{and} \quad v_0(z) = c_0 + \sum_{k=1}^{h_c} c_k(z^k + z^{-k}),$$

where h_b, h_c are nonnegative integers and $b_j, c_k, j = 0, \ldots, h_b, k = 0, \ldots, h_c$ are real numbers which are to be determined later. Let us consider the following four cases.

Case 1. $[SB](z)[Sd](z) = z^{2n}$ for some $n \in \mathbb{Z}$. We choose $u_1(z) = z^n u_0(z)$ and $v_1(z) = v_0(z)$. When N is even, set $h_b = h_c = N/2$; when N is odd, set $h_b = h_c = (N-1)/2$.

Case 2. $[SB](z)[Sd](z) = z^{2n+1}$ for some $n \in \mathbb{Z}$. We choose $u_1(z) = z^n(1+z)u_0(z)$ and $v_1(z) = v_0(z)$. When N is even, set $h_b = N/2 - 1$ and $h_c = N/2$; when N is odd, set $h_b = h_c = (N-1)/2$.

Case 3. $[SB](z)[Sd](z) = -z^{2n}$ for some $n \in \mathbb{Z}$. When N is even, we choose $u_1(z) = z^n(z - 1/z)u_0(z)$, $v_1(z) = v_0(z)$ and set $h_b = N/2 - 1, h_c = N/2$; when N

is odd, we choose $u_1(z) = z^n(1-z)u_0(z)$, $v_1(z) = (1+1/z)v_0(z)$ and set $h_b = h_c = (N-1)/2$.

Case 4. $[SB](z)[Sd](z) = -z^{2n+1}$ for some $n \in \mathbb{Z}$. We choose $u_1(z) = z^n(1-z)u_0(z)$ and $v_1(z) = v_0(z)$. When N is even, set $h_b = N/2-1$ and $h_c = N/2$; when N is odd, set $h_b = h_c = (N-1)/2$.

It is easy to see that both u_1 and v_1 are (anti)symmetric and (4.5) holds. Moreover, the degrees of u_1 and v_1 are at most N and it is easy to verify that $h_b+h_c+2 > N$. Since $A(z) > 0$ for all $z \in \mathbb{T}$, by the Fejér–Riesz lemma, we have $A(z) = \tilde{A}(z)\tilde{A}(1/z)$ for some Laurent polynomial $\tilde{A}$ with real coefficients such that all of the roots of $\tilde{A}$ are contained in $\{z \in \mathbb{C} : |z| < 1\}$. Therefore, $\tilde{A}(z)$ and $\tilde{A}(1/z)$ have no common zeros in $\mathbb{C}\backslash\{0\}$. Since $A(z) = A_0 + \sum_{k=1}^{N} A_k(z^k + z^{-k})$, $\tilde{A}(z)$ can have at most N zeros in $\mathbb{C}\backslash\{0\}$, say, $\{z_1, \dots, z_{N'}\}$ are all of the distinct roots of the Laurent polynomial $\tilde{A}(z)$ in $\mathbb{C}\backslash\{0\}$ such that $Z(\tilde{A}, z_1) + \cdots + Z(\tilde{A}, z_{N'}) = N$. Define $F(z) := B(1/z)u_1(z) - d(z)v_1(1/z)$. Now we have the following system of homogeneous linear equations:

$$F^{(j)}(z_k) = 0, \qquad k = 0, \dots, N',\ j = 0, \dots, Z(\tilde{A}, z_k) - 1. \tag{4.6}$$

Since the number of free parameters in $\{c_j, d_k : j = 0, \dots, h_b, k = 0, \dots, h_c\}$ is $h_b + h_c + 2 > N$ and we have N homogeneous linear equations, there must be a nonzero solution $\{c_j, d_k : j = 0, \dots, h_b, k = 0, \dots, h_c\}$ to the system of homogeneous linear equations in (4.6). So there exist u_1 and v_1 satisfying (4.6) with at least one of them nonzero. In other words, we deduce from (4.6) that

$$\tilde{A}(z) \mid [B(1/z)u_1(z) - d(z)v_1(1/z)]. \tag{4.7}$$

Since $z_1, \dots, z_{N'}$ are complex numbers, a solution $\{c_j, d_k : j = 0, \dots, h_b, k = 0, \dots, h_c\}$ may be complex numbers too. However, since $\tilde{A}, B$, and C are Laurent polynomials with real coefficients, we can simply replace the numbers c_j, d_k by either their real parts or their imaginary part so that (4.7) is still true and at least one of u_1 and v_1 is nonzero.

On the other hand, by (4.5) and Proposition 2.1, we deduce that $B(1/z)u_1(z) - d(z)v_1(1/z)$ is (anti)symmetric. So,

$$B(z)u_1(1/z) - d(1/z)v_1(z) = p(z)[B(1/z)u_1(z) - d(z)v_1(1/z)] \tag{4.8}$$

for some nonzero trivial Laurent polynomial p. Consequently, it follows from (4.7) and (4.8) that

$$\tilde{A}(1/z) \mid [B(1/z)u_1(z) - d(z)v_1(1/z)].$$

Since $\tilde{A}(z)$ and $\tilde{A}(1/z)$ have no common zeros in $\mathbb{C}\backslash\{0\}$ and $A(z) = \tilde{A}(z)\tilde{A}(1/z)$, we conclude that (4.4) holds. Later on we shall show that $u_1(1)^2 + v_1(1)^2 \neq 0$. If $u_1(1)^2 + v_1(1)^2 \neq 0$, then we can properly scale u_1 and v_1 such that $u_1(1)^2 + v_1(1)^2 = A(1)$ holds. Note that without factorizing A we can solve the system of linear equations given by $[B(1/z)u_1(z) - d(z)v_1(1/z)] \equiv 0 \mod A(z)$ to obtain the desired u_1 and v_1.

Since $A(z) \not\equiv 0$, we can define

$$u_2(z) := \frac{B(1/z)u_1(z) - d(z)v_1(1/z)}{A(z)} \quad \text{and} \quad v_2(z) := \frac{d(z)u_1(1/z) + B(1/z)v_1(z)}{A(z)}. \tag{4.9}$$

By (4.4) we see that u_2 is an (anti)symmetric Laurent polynomial with real coefficients. Now we show that v_2 is also an (anti)symmetric Laurent polynomial. By definition of u_2 and the fact that $d(z)d(1/z) = \det M(z) = A(z)C(z) - B(z)B(1/z)$, we have

$$\begin{aligned} A(z)d(1/z)u_2(z) &= B(1/z)d(1/z)u_1(z) - \det M(z)v_1(1/z) \\ &= B(1/z)d(1/z)u_1(z) - A(z)C(z)v_1(1/z) + B(1/z)B(z)v_1(1/z). \end{aligned}$$

From the above identity, we have

$$A(z)[d(1/z)u_2(z) + C(z)v_1(1/z)] = B(1/z)[d(1/z)u_1(z) + B(z)v_1(1/z)].$$

Since $A(z) = A(1/z)$ and $\gcd(A, B) = 1$, we conclude that $A(z) \mid [d(1/z)u_1(z) + B(z)v_1(1/z)]$ and therefore, by $A(1/z) = A(z)$, $A(z) \mid [d(z)u_1(1/z) + B(1/z)v_1(z)]$. So v_2 is a Laurent polynomial with real coefficients. By (4.5) and Proposition 2.1, we see that v_2 is (anti)symmetric.

By (4.9) and Proposition 2.1, we see that (2.7) and the system of linear equations in (4.1) must hold. In the following, let us show that $u_1(1)^2 + v_1(1)^2 \neq 0$. Since both (2.7) and (4.1) are satisfied, as we demonstrated in the first part of the proof, we must have $u_1(z)u_1(1/z) + v_1(z)v_1(1/z) = pA(z)$ for some constant p. If $u_1(1) = v_1(1) = 0$, by $A(1) > 0$, then we must have $p = 0$. That is, $|u_1(z)|^2 + |v_1(z)|^2 = u_1(z)u_1(1/z) + v_1(z)v_1(1/z) = 0$ for all $z \in \mathbb{T}$. So, u_1 and v_1 must be identically zero which is a contradiction to our choice of u_1 and v_1 since one of them must be nonzero. So $u_1(1)^2 + v_1(1)^2 \neq 0$. Now replacing u_1 and v_1 by cu_1 and cv_1 with $c = \sqrt{A(1)/(u_1(1)^2 + v_1(1)^2)}$ in the above proof, we see that (4.1) and (2.7) still hold. Moreover, we have $u_1(1)^2 + v_1(1)^2 = A(1)$ which completes the proof. □

Let $\mathbb{R}[z, z^{-1}]$ denote all of the Laurent polynomials with real coefficients. For a Laurent polynomial $p \in \mathbb{R}[z, z^{-1}]$, we say that p is *irreducible* in $\mathbb{R}[z, z^{-1}]$ if $q \mid p$ for some $q \in \mathbb{R}[z, z^{-1}]$ implies that $q = p_0$ or $q = p_0 p$ for some trivial Laurent polynomial $p_0 \in \mathbb{R}[z, z^{-1}]$ (that is, $p_0 = cz^k$ for some $c \in \mathbb{R}\backslash\{0\}$ and $k \in \mathbb{Z}$).

Now we have a stronger version of Theorem 4.1.

COROLLARY 4.2. *Let A, B, and C be (anti)symmetric Laurent polynomials with real coefficients. Let M be defined in* (2.4)*. Suppose that $M(z) \geqslant 0$ for all $z \in \mathbb{T}$ and $\det M(z) = d(z)d(1/z)$ for some (anti)symmetric Laurent polynomial d with real coefficients. If $\gcd(A, B, C) = 1$, then there exist four (anti)symmetric Laurent polynomials u_1, u_2, v_1, v_2 with real coefficients such that* (2.5) *and* (2.7) *are satisfied.*

Proof. If $C(z) \equiv 0$, then $\gcd(A, B) = \gcd(A, B, C) = 1$ and all the claims follow from Theorem 4.1. So, we can assume that C is not identically zero.

Define $h(z) = \gcd(A(z), B(z)B(1/z))$. By the symmetry of A and B, we see that h must be (anti)symmetric. Now, we show that $\gcd(h, C) = 1$. Suppose not. Then there is a nontrivial irreducible $p \in \mathbb{R}[z, z^{-1}]$ such that $p \mid \gcd(h, C)$. So, $p \mid h$ and $p \mid C$. Consequently, $p \mid A$ and $p \mid B(z)B(1/z)$. Note that $B(1/z) = B(z)/[SB](z)$ and SB is trivial. So $p \mid B^2$. Since p is irreducible, we must have $p \mid B$. So, $p \mid \gcd(A, B, C)$ which is a contradiction since p is nontrivial but by assumption $\gcd(A, B, C) = 1$.

Next, we show that for a nontrivial irreducible $p \in \mathbb{R}[z, z^{-1}]$, if $p^{2n-1} \mid h$ for some $n \in \mathbb{N}$, then we must have $p^{2n} \mid h$. Since $p^{2n-1} \mid h$, we have $p^{2n-1} \mid B(z)B(1/z)$ and therefore, $p^{2n-1} \mid B^2$. Since p is irreducible, we must have $p^n \mid B$ and consequently $p^{2n} \mid B(z)B(1/z)$.

On the other hand, by $p^{2n-1} \mid h$ and $h = \gcd(A(z), B(z)B(1/z))$, we have

$$p^{2n-1} \mid [A(z)C(z) - B(z)B(1/z)].$$

Since $A(z)C(z) - B(z)B(1/z) = \det M(z) = d(z)d(1/z)$, we have $p^{2n-1} \mid d(z)d(1/z)$. Since $d(1/z) = d(z)/[Sd](z)$ and Sd is trivial, it follows from $p^{2n-1} \mid d^2$ that $p^{2n} \mid d(z)d(1/z)$. Since $C \not\equiv 0$, by $d(z)d(1/z) = \det M(z) = A(z)C(z) - B(z)B(1/z)$, we have

$$A(z) = \frac{d(z)d(1/z) + B(z)B(1/z)}{C(z)}.$$

By $\gcd(h, C) = 1$ and $p \mid h$, we must have $\gcd(p, C) = 1$ since p is nontrivial irreducible. Hence, we must have $p^{2n} \mid A$. So, $p^{2n} \mid h$. As a consequence of the fact that $p^{2n-1} \mid h$ implies $p^{2n} \mid h$, factorize h as

$$h(z) = p_0(z) \prod_{j=1}^{m} p_j^{2n_j}(z),$$

where p_0 is a trivial Laurent polynomial and $p_1, \dots, p_m$ are essentially different nontrivial irreducible Laurent polynomials in $\mathbb{R}[z, z^{-1}]$. Now define

$$d_h(z) := \prod_{j=1}^{m} p_j^{n_j}(z).$$

Then $h(z) = p_0(z)d_h(z)d_h(z)$. Note that by Proposition 2.2 we can directly obtain d_h from h without factorizing h. Since $([Sd_h](z))^2 = [Sh](z)/[Sp_0](z)$ is a trivial Laurent polynomial, Sd_h must be trivial and therefore d_h is (anti)symmetric. So, $\gcd(A(z), B(z)B(1/z)) = d_h(z)d_h(1/z)$. Since both d_h and B are (anti)symmetric, it follows from $d_h(z)d_h(1/z) \mid B(z)B(1/z)$ that $d_h^2 \mid B^2$ and consequently $d_h \mid B$. Define

$$\tilde{A}(z) := \frac{A(z)}{d_h(z)d_h(1/z)}, \quad \tilde{B}(z) := \frac{B(z)}{d_h(z)} \quad \text{and} \quad \tilde{M}(z) = \begin{bmatrix} \tilde{A}(z) & \tilde{B}(z) \\ \tilde{B}(1/z) & C(z) \end{bmatrix}.$$

Clearly, $\tilde{A}, \tilde{B}$, and C are (anti)symmetric Laurent polynomials and $\gcd(\tilde{A}, \tilde{B}) = 1$. By Theorem 4.1, there exist four (anti)symmetric Laurent polynomials $\tilde{u}_1, \tilde{u}_2, \tilde{v}_1, \tilde{v}_2$ with real coefficients such that

$$\tilde{M}(z) = \begin{bmatrix} \tilde{u}_1(z) & \tilde{v}_1(z) \\ \tilde{u}_2(z) & \tilde{v}_2(z) \end{bmatrix} \begin{bmatrix} \tilde{u}_1(1/z) & \tilde{u}_2(1/z) \\ \tilde{v}_1(1/z) & \tilde{v}_2(1/z) \end{bmatrix} \tag{4.10}$$

and

$$\frac{[S\tilde{u}_1](z)}{[S\tilde{u}_2](z)} = [S\tilde{M}_{1,2}](z) = \frac{[S\tilde{v}_1](z)}{[S\tilde{v}_2](z)}. \tag{4.11}$$

Note that

$$M(z) = \begin{bmatrix} d_h(z) & 0 \\ 0 & 1 \end{bmatrix} \tilde{M}(z) \begin{bmatrix} d_h(1/z) & 0 \\ 0 & 1 \end{bmatrix}.$$

Define

$$u_1(z) = \tilde{u}_1(z)d_h(z), \quad v_1(z) = \tilde{v}_1(z)d_h(z), \quad u_2(z) = \tilde{u}_2(z), \quad v_2(z) = \tilde{v}_2(z).$$

Then it follows directly from (4.10) and (4.11) that (2.5) and (2.7) are satisfied. ∎

LEMMA 4.3. *Let p be a nonzero (anti)symmetric Laurent polynomial with real coefficients. Then there exist $c \in \{-1, 1\}$ and $k \in \mathbb{Z}$ such that $cz^k p(z) \geqslant 0$ for all $z \in \mathbb{T}$ if and only if $Z(p, z_0)$ is an even integer for every $z_0 \in \mathbb{T}$.*

Proof. If $cz^k p(z) \geqslant 0$ for all $z \in \mathbb{T}$, then by the Fejér–Riesz lemma, $cz^k p(z) = q(z)q(1/z)$ for some Laurent polynomial q with real coefficients. Hence for all $z_0 \in \mathbb{T}$, we have

$$Z(p, z_0) = Z(cz^k p(z), z_0) = Z(q(z), z_0) + Z(q(1/z), z_0) = 2Z(q(z), z_0),$$

where we used the fact that $Z(q(1/z), z_0) = Z(\overline{q(z)}, z_0) = Z(q, z_0)$ for all $z_0 \in \mathbb{T}$ since q is a Laurent polynomial with real coefficients. So $Z(p, z_0)$ must be an even integer for every $z_0 \in \mathbb{T}$.

Conversely, write $p(z) = q(z)h(z)$ such that $q(z) \neq 0$ for all $z \in \mathbb{T}$ and all of the zeros of h lie on $\mathbb{T}$. Since p is (anti)symmetric and $Z(p, z_0) = Z(h, z_0)$ is an even integer for all $z_0 \in \mathbb{T}$, there exist $c_1 \in \{-1, 1\}$ and $k_1 \in \mathbb{Z}$ such that $c_1 z^{k_1} h(z) \geqslant 0$ for all $z \in \mathbb{T}$. Since $[Sp](z) = [Sq](z)[Sh](z)$, q must be symmetric. Since $q(z) \neq 0$ for all $z \in \mathbb{T}$, we must have $[Sq](z) = z^{2k_2}$ for some $k_2 \in \mathbb{Z}$. So $z^{-k_2} q(z) \neq 0$ and is real-valued for all $z \in \mathbb{T}$. Consequently, there exists $c_2 \in \{-1, 1\}$ such that $c_2 z^{-k_2} q(z) > 0$ for all $z \in \mathbb{T}$. So, $c_1 c_2 z^{k_1 - k_2} p(z) = c_1 z^{k_1} h(z) c_2 z^{-k_2} q(z) \geqslant 0$ for all $z \in \mathbb{T}$. □

When p is antisymmetric, it is evident that the condition in Lemma 4.3 cannot be satisfied.

LEMMA 4.4. *Let g be a nonzero Laurent polynomial with real coefficients. Then there exist two (anti)symmetric Laurent polynomials q_1 and q_2 with real coefficients such that*

$$q_1(z)q_1(1/z) + q_2(z)q_2(1/z) = g(z) \tag{4.12}$$

and

$$[Sq_1](z)/[Sq_2](z) = z^{2k}, -z^{2k}, z^{2k+1}, \text{ or } -z^{2k+1} \qquad \text{for some integer } k$$

if and only if $g(z) \geqslant 0$ for all $z \in \mathbb{T}$ and $Z(g, x)$ is an even integer for every $x \in (-1, 0) \cup (0, 1)$, $x \in \emptyset$, $x \in (0, 1)$, or $x \in (-1, 0)$, respectively.

Proof. Necessity. If (4.12) holds, then it is evident that $g(z) \geqslant 0$ for all $z \in \mathbb{T}$. Since $q_1(1/z) = q_1(z)/[Sq_1](z)$ and $q_2(1/z) = q_2(z)/[Sq_2](z)$, we can rewrite (4.12) as follows:

$$q_1^2(z) + q_2^2(z)[Sq_1](z)/[Sq_2](z) = g(z)[Sq_1](z).$$

If $[Sq_1](z)/[Sq_2](z) = z^{2k}$, then we have $q_1^2(x) + x^{2k} q_2^2(x) = g(x)[Sq_1](x)$ for all $x \in \mathbb{R}\backslash\{0\}$ and consequently, it is easy to see that for every $x \in (-1, 0) \cup (0, 1)$, we have

$$Z(g, x) = Z(g[Sq_1], x) = \min(Z(q_1^2, x), Z(q_2^2, x)) = 2\min(Z(q_1, x), Z(q_2, x)).$$

So, when $[Sq_1](z)/[Sq_2](z) = z^{2k}$, $Z(g, x)$ must be an even integer for all $x \in (-1, 0) \cup (0, 1)$.

If $[Sq_1](z)/[Sq_2](z) = z^{2k+1}$, then we have $q_1^2(x) + x^{2k+1} q_2^2(x) = g(x)[Sq_1](x)$ for all $x \in \mathbb{R}\backslash\{0\}$. Similarly, it is easy to prove that $Z(g, x) = 2\min(Z(q_1, x), Z(q_2, x))$ must be an even integer for every $x \in (0, 1)$.

If $[Sq_1](z)/[Sq_2](z) = -z^{2k+1}$, then we have $q_1^2(x) + (-x)^{2k+1} q_2^2(x) = g(x)[Sq_1](x)$ for all $x \in \mathbb{R}\backslash\{0\}$. Similarly, it is easy to prove that $Z(g, x) = 2\min(Z(q_1, x), Z(q_2, x))$ must be an even integer for every $x \in (-1, 0)$.

Sufficiency. Since $g(z) \geqslant 0$ for all $z \in \mathbb{T}$, by the Fejér–Riesz lemma, we can write $g(z) = h(z)h(1/z)$ for some Laurent polynomial h with real coefficients such that all of the roots of h are contained in $\{z : |z| \leqslant 1\}$. Set $q_1(z) = z^k[h(z) + h(1/z)]/2$ and $q_2(z) = [h(z) - h(1/z)]/2$. Then it is easy to verify that (4.12) holds and $[Sq_1](z)/[Sq_2](z) = -z^{2k}$. In the following, let us consider the other three cases. Factorize h as

$$h(z) = p_0(z)(z-1)^{Z(h,1)}(z+1)^{Z(h,-1)} \prod_{j=1}^{m} p_j^{n_j}(z),$$

where p_0 is a trivial Laurent polynomial and all $p_j, j = 1, \dots, m$ are essentially different nontrivial irreducible Laurent polynomials in $\mathbb{R}[z, z^{-1}]$. Since there are only two types of nontrivial irreducible Laurent polynomials in $\mathbb{R}[z, z^{-1}]$, without lost of generality we can assume that either $p_j = z - a_j$ for some $a_j \in (-1, 1)\backslash\{0\}$ or $p_j(z) = z^2 + b_j z + c_j$ for some $b_j, c_j \in \mathbb{R}$ satisfying $b_j^2 - 4c_j < 0$. Let us consider the following two cases.

If $p_j(z) = z^2 + b_j z + c_j$ for some $b_j, c_j \in \mathbb{R}$ satisfying $4c_j > b_j^2$, since $c_j \geqslant 0$ and $-2\sqrt{c_j} \leqslant b_j \leqslant 2\sqrt{c_j}$, then we have

$$\begin{aligned} p_j(z) &= [z + b_j/2]^2 + z^{2k}\left[\sqrt{c_j - b_j^2/4}\, z^{-k}\right]^2 \\ &= [z - \sqrt{c_j}]^2 + z^{2k+1}\left[\sqrt{2\sqrt{c_j} + b_j}\, z^{-k}\right]^2 \\ &= [z + \sqrt{c_j}]^2 - z^{2k+1}\left[\sqrt{2\sqrt{c_j} - b_j}\, z^{-k}\right]^2. \end{aligned}$$

If $p_j(z) = z - a_j$ for some $a_j \in (-1, 1)\backslash\{0\}$, then by assumption, we have the following cases:

Case 1: If $Z(g, x)$ is an even integer for all $x \in (-1, 0) \cup (0, 1)$, then n_j must be an even integer and therefore, $p_j^{n_j}(z) = [(z - a_j)^{n_j/2}]^2 + z^{2k} \times 0$.

Case 2: If $Z(g, x)$ is an even integer for all $x \in (0, 1)$, then n_j must be an even integer when $a_j \in (0, 1)$. Therefore, when $a_j \in (0, 1)$, we have $p_j^{n_j}(z) = [(z-a_j)^{n_j/2}]^2 + z^{2k+1} \times 0$. When $a_j \in (-1, 0)$, we also have $p_j(z) = z - a_j = [\sqrt{-a_j}]^2 + z^{2k+1}[z^{-k}]^2$.

Case 3: If $Z(g, x)$ is an even integer for all $x \in (-1, 0)$, then n_j must be an even integer if $a_j \in (-1, 0)$. When $a_j \in (-1, 0)$, we have $p_j^{n_j}(z) = [(z-a_j)^{n_j/2}]^2 - z^{2k+1} \times 0$. When $a_j \in (0, 1)$, we also have $p_j(z) = z - a_j = -\big([\sqrt{a_j}]^2 - z^{2k+1}[z^{-k}]^2\big)$.

By a direct computation, it is easy to verify the following identity:

$$(f_1^2 + wf_2^2)(f_3^2 + wf_4^2) = (f_1 f_3 - wf_2 f_4)^2 + w(f_1 f_4 + f_2 f_3)^2. \tag{4.13}$$

By the above argument, using the identity in (4.13) we have

$$h(z) = \tilde{q}_0(z)(z-1)^{Z(h,1)}(z+1)^{Z(h,-1)}\big(\tilde{q}_1^2(z) + w(z)\tilde{q}_2^2(z)\big),$$

where $\tilde{q}_0$ is a trivial Laurent polynomial, $w(z) = z^{2k}, z^{2k+1}$, or $-z^{2k+1}$ according to the assumption, and $\tilde{q}_1$ and $\tilde{q}_2$ are Laurent polynomials with real coefficients. Observing that $w(1/z) = w(z)^{-1}$, we have

$$h(1/z) = \tilde{q}_0(1/z)(1/z-1)^{Z(h,1)}(1/z+1)^{Z(h,-1)}\big(\tilde{q}_1^2(1/z) + w(z)[\tilde{q}_2(1/z)/w(z)]^2\big).$$

Note that $\tilde{q}_0(z)\tilde{q}_0(1/z)$ is a positive constant since $\tilde{q}_0$ is trivial. By a simple computation, we deduce that

$$g(z) = h(z)h(1/z) = q_1(z)q_1(1/z) + q_2(z)q_2(1/z),$$

where

$$q_1(z):=\sqrt{\tilde{q}_0(z)\tilde{q}_0(1/z)}(z-1)^{Z(h,1)}(z+1)^{Z(h,-1)}[\tilde{q}_1(z)\tilde{q}_1(1/z)-\tilde{q}_2(z)\tilde{q}_2(1/z)],$$
$$q_2(z):=\sqrt{\tilde{q}_0(z)\tilde{q}_0(1/z)}(z-1)^{Z(h,1)}(z+1)^{Z(h,-1)}[\tilde{q}_1(z)\tilde{q}_2(1/z)w(z)^{-1}+\tilde{q}_2(z)\tilde{q}_1(1/z)].$$

Since $w(z)^{-1}=w(1/z)$, by a simple computation, we have

$$q_1(1/z)=(-1)^{Z(h,1)}z^{-Z(h,1)-Z(h,-1)}q_1(z),$$
$$q_2(1/z)=(-1)^{Z(h,1)}z^{-Z(h,1)-Z(h,-1)}w(z)q_2(z).$$

Therefore, both q_1 and q_2 are (anti)symmetric and $[Sq_1](z)/[Sq_2](z)=w(z)$. ∎

5. Proof of Theorem 2.3 and its associated algorithm. In this section, we shall prove Theorem 2.3 and give a step-by-step algorithm to implement it.

Proof of Theorem 2.3. If $g=\gcd(A,B,C)\equiv 0$, then $A=B=C=0$ and all the claims are obviously true by taking $u_1=u_2=v_1=v_2=0$. So, we will assume $g\not\equiv 0$.

Since $g=\gcd(A,B,C)$, by the symmetry of A,B, and C, g is (anti)symmetric. Since $\det M(z)\geqslant 0$ for all $z\in\mathbb{T}$, we see that

$$0\leqslant B(z)B(1/z)\leqslant A(z)C(z)\qquad\forall\, z\in\mathbb{T}.$$

Since $B(1/z)=B(z)/[SB](z)$, it yields that $2Z(B,z)\geqslant Z(A,z)+Z(C,z)$ for all $z\in\mathbb{T}$. So, by the definition of g, we have $Z(g,z)=\min(Z(A,z),Z(B,z),Z(C,z))=\min(Z(A,z),Z(C,z))$ for every $z\in\mathbb{T}$. Since $A(z)\geqslant 0$ and $C(z)\geqslant 0$ for all $z\in\mathbb{T}$, by Lemma 4.3, $Z(A,z)$ and $Z(C,z)$ are even integers. Consequently, $Z(g,z)=\min(Z(A,z),Z(C,z))$ is an even integer for all $z\in\mathbb{T}$. Since g is (anti)symmetric, by Lemma 4.3, there exist $c\in\{-1,1\}$ and $k\in\mathbb{Z}$ such that $cz^kg(z)\geqslant 0$ for all $z\in\mathbb{T}$. Since $g=\gcd(A,B,C)$, without loss of generality, we can assume that $g(z)\geqslant 0$ for all $z\in\mathbb{T}$ by replacing g by $cz^kg(z)$. Now define $\tilde{M}(z)=M(z)/g(z)$ by

$$\tilde{M}(z)=\begin{bmatrix}\tilde{A}(z) & \tilde{B}(z)\\ \tilde{B}(1/z) & \tilde{C}(z)\end{bmatrix}\quad\text{with}$$
$$\tilde{A}(z)=\frac{A(z)}{g(z)},\quad \tilde{B}(z)=\frac{B(z)}{g(z)},\quad \tilde{C}(z)=\frac{C(z)}{g(z)}. \tag{5.1}$$

Since $g(z)\geqslant 0$ for all $z\in\mathbb{T}$, it is easy to see that all $\tilde{A},\tilde{B},\tilde{C}$ are (anti)symmetric Laurent polynomials and $\tilde{M}(z)\geqslant 0$ for all $z\in\mathbb{T}$.

Sufficiency. Since $d(z)d(1/z)=\det M(z)=g(z)^2\det\tilde{M}(z)$, we have $g^2\mid d(z)d(1/z)$. Since $d(1/z)=d(z)/[Sd](z)$, $g^2\mid d^2$ and therefore, $g\mid d$. So define $d_1(z)=d(z)/g(z)$. Then d_1 is an (anti)symmetric Laurent polynomial and $\det\tilde{M}(z)=d_1(z)d_1(1/z)$. Note that $\gcd(\tilde{A},\tilde{B},\tilde{C})=1$. By Corollary 4.2, there exist four (anti)symmetric Laurent polynomials $\tilde{u}_1,\tilde{u}_2,\tilde{v}_1,\tilde{v}_2$ with real coefficients such that (4.10) and (4.11) are satisfied. Define

$$\tilde{d}(z):=\tilde{u}_1(z)\tilde{v}_2(z)-\tilde{u}_2(z)\tilde{v}_1(z).$$

By (4.11), $\tilde{d}$ is (anti)symmetric and by Proposition 2.1 $[S\tilde{d}](z)=[S\tilde{u}_1](z)[S\tilde{v}_2](z)$.

By Proposition 2.2, it follows from $\tilde{d}(z)\tilde{d}(1/z)=\det\tilde{M}(z)=d_1(z)d_1(1/z)$ that we must have $\tilde{d}(z)=\pm z^kd_1(z)=\pm z^kd(z)/g(z)$ for some $k\in\mathbb{Z}$. So, $[S\tilde{d}](z)=z^{2k}[Sd](z)$. Rewrite (4.11) as

$$\frac{[S\tilde{u}_1](z)}{[S\tilde{u}_2](z)}=[S\tilde{B}](z)=[SB](z)=\frac{[S\tilde{v}_1](z)}{[S\tilde{v}_2](z)}.$$

So, we have

$$\frac{[S\tilde{v}_1](z)}{[S\tilde{u}_1](z)} = \frac{[S\tilde{v}_1](z)}{[S\tilde{v}_2](z)}\frac{[S\tilde{u}_1](z)[S\tilde{v}_2](z)}{([S\tilde{u}_1](z))^2} = [SB](z)\frac{[S\tilde{d}](z)}{([S\tilde{u}_1](z))^2}$$
$$= \left(\frac{z^k}{[S\tilde{u}_1](z)}\right)^2 [SB](z)[Sd](z)$$

and

$$\frac{[S\tilde{v}_2](z)}{[S\tilde{u}_2](z)} = \frac{[S\tilde{u}_1](z)}{[S\tilde{u}_2](z)}\frac{[S\tilde{u}_1](z)[S\tilde{v}_2](z)}{(S\tilde{u}_1](z))^2} = [SB](z)\frac{[S\tilde{d}](z)}{([S\tilde{u}_1](z))^2} = \frac{[S\tilde{v}_1](z)}{[S\tilde{u}_1](z)}.$$

By assumption in (c) and Lemma 4.4, there exist two (anti)symmetric Laurent polynomials q_1 and q_2 such that

$$\frac{[Sq_1](z)}{[Sq_2](z)} = \frac{[S\tilde{v}_1](z)}{[S\tilde{u}_1](z)} = \frac{[S\tilde{v}_2](z)}{[S\tilde{u}_2](z)} = \left(\frac{z^k}{[S\tilde{u}_1](z)}\right)^2 [SB](z)[Sd](z) \tag{5.2}$$

and $g(z) = q_1(z)q_1(1/z) + q_2(z)q_2(1/z)$. Define

$$\begin{bmatrix} u_1(z) & v_1(z) \\ u_2(z) & v_2(z) \end{bmatrix} = \begin{bmatrix} \tilde{u}_1(z) & \tilde{v}_1(z) \\ \tilde{u}_2(z) & \tilde{v}_2(z) \end{bmatrix} \begin{bmatrix} q_1(z) & -q_2(1/z) \\ q_2(z) & q_1(1/z) \end{bmatrix}.$$

Now by (5.2) and Proposition 2.1, it is easy to check that all u_1, u_2, v_1, v_2 are (anti)symmetric Laurent polynomials. By a direct computation, it is easy to see that (2.5) and (2.7) are satisfied.

Necessity. Obviously, (a) and (b) must be true. We shall prove that (c) must be true. We can assume that $d \not\equiv 0$. As we proved before the part of sufficiency, we can assume that $g(z) \geqslant 0$ for all $z \in \mathbb{T}$. Let $\tilde{M}$ be defined in (5.1). We have $g(z)^2 \det\tilde{M}(z) = \det M(z) = d(z)d(1/z)$. So, $g^2 \mid d(z)d(1/z)$. Since $d(1/z) = d(z)/[Sd](z)$, we deduce that $g^2 \mid d^2$ and therefore, $g \mid d$. Define $\tilde{d}(z) = d(z)/g(z)$. Then $\tilde{d}$ is (anti)symmetric and $\det\tilde{M}(z) = \tilde{d}(z)\tilde{d}(1/z)$. Since $M(z) \geqslant 0$ and $g(z) \geqslant 0$ for all $z \in \mathbb{T}$, it is easy to see that $\tilde{M}(z) \geqslant 0$ for all $z \in \mathbb{T}$. Since $\gcd(\tilde{A}, \tilde{B}, \tilde{C}) = 1$, by Corollary 4.2, (4.10) and (4.11) are satisfied. So,

$$\begin{bmatrix} u_1(z) & v_1(z) \\ u_2(z) & v_2(z) \end{bmatrix} \begin{bmatrix} u_1(1/z) & u_2(1/z) \\ v_1(1/z) & v_2(1/z) \end{bmatrix} = g(z)\tilde{M}(z)$$
$$= g(z) \begin{bmatrix} \tilde{u}_1(z) & \tilde{v}_1(z) \\ \tilde{u}_2(z) & \tilde{v}_2(z) \end{bmatrix} \begin{bmatrix} \tilde{u}_1(1/z) & \tilde{u}_2(1/z) \\ \tilde{v}_1(1/z) & \tilde{v}_2(1/z) \end{bmatrix}.$$

Define

$$Q(z) := \begin{bmatrix} q_1(z) & q_2(z) \\ q_3(z) & q_4(z) \end{bmatrix} := \begin{bmatrix} \tilde{v}_2(z) & -\tilde{v}_1(z) \\ -\tilde{u}_2(z) & \tilde{u}_1(z) \end{bmatrix} \begin{bmatrix} u_1(z) & v_1(z) \\ u_2(z) & v_2(z) \end{bmatrix}.$$

Then $Q(z)Q(1/z)^T = g(z)\tilde{d}(z)\tilde{d}(1/z)I_2$. In particular, we have

$$q_1(z)q_1(1/z) + q_2(z)q_2(1/z) = g(z)\tilde{d}(z)\tilde{d}(1/z).$$

By (2.7) and (4.11), we have

$$\frac{[Su_1](z)}{[Su_2](z)} = \frac{[Sv_1](z)}{[Sv_2](z)} = [SB](z) = [S\tilde{B}](z) = \frac{[S\tilde{u}_1](z)}{[S\tilde{u}_2](z)} = \frac{[S\tilde{v}_1](z)}{[S\tilde{v}_2](z)}.$$

By Proposition 2.1, q_1 and q_2 are (anti)symmetric. By Proposition 2.2, $d(z) = \pm z^k[u_1(z)v_2(z) - u_2(z)v_1(z)]$. So,

$$[Sd](z) = z^{2k}[Su_1](z)[Sv_2](z) = z^{2k}[Su_2](z)[Sv_1](z).$$

Observing that

$$[Sq_1](z) = [S\tilde{v}_2](z)[Su_1](z) \quad \text{and} \quad [Sq_2](z) = [S\tilde{v}_2](z)[Sv_1](z),$$

we have

$$\frac{[Sq_1](z)}{[Sq_2](z)} = \frac{[S\tilde{v}_2](z)[Su_1](z)}{[S\tilde{v}_2](z)[Sv_1](z)} = \frac{[Su_1](z)}{[Su_2](z)} \frac{z^{2k}[Su_2](z)[Sv_1](z)}{(z^k[Sv_1](z))^2} = \frac{[SB](z)[Sd](z)}{(z^k[Sv_1](z))^2}.$$

By Lemma 4.4, $Z(g(z)\tilde{d}(z)\tilde{d}(1/z), x)$ must be an even integer for the corresponding cases. Note that $\tilde{d}(1/z) = \tilde{d}(z)/[S\tilde{d}](z)$. So, $Z(\tilde{d}(z)\tilde{d}(1/z), x)$ is always an even integer for all $x \in \mathbb{R}$. So, $Z(g, x) = Z(g(z)\tilde{d}(z)\tilde{d}(1/z), x) - Z(\tilde{d}(z)\tilde{d}(1/z), x)$ must be an even integer for the corresponding cases. Therefore, (c) must be true.

Finally, by the proof of Theorem 2.3 and all the auxiliary results in section 4, let us present the following algorithm on splitting a matrix of Laurent polynomials with symmetry.

ALGORITHM 5.1. *Let A, B, and C be (anti)symmetric Laurent polynomials with real coefficients. Let M be the 2×2 matrix defined in* (2.4) *such that all the conditions in Theorem* 2.3 *are satisfied.*

(1) *Compute $g = gcd(A, B, C)$. By the proof of Theorem* 2.3*, without loss of generality, we can assume that $g \not\equiv 0$ and $g(z) \geqslant 0$ for all $z \in \mathbb{T}$.*

(2) *Compute $h(z) = gcd(A(z)/g(z), B(z)B(1/z)/g(z)^2)$. By the proof of Corollary* 4.2*, we can assume that $h(z) \geqslant 0$ for all $z \in \mathbb{T}$ and we can calculate d_h such that $h(z) = d_h(z)d_h(1/z)$ by Proposition* 2.2*.*

(3) *Define a 2×2 matrix $\tilde{M}$ of Laurent polynomials with real coefficients by*

$$\tilde{M}(z) = \begin{bmatrix} \tilde{A}(z) & \tilde{B}(z) \\ \tilde{B}(1/z) & \tilde{C}(z) \end{bmatrix}$$

with

$$\tilde{A}(z) = \frac{A(z)}{g(z)h(z)}, \quad \tilde{B}(z) = \frac{B(z)}{g(z)d_h(z)}, \quad \tilde{C}(z) = \frac{C(z)}{g(z)}.$$

By Proposition 2.2*, we can calculate $\tilde{d}$ such that $\det\tilde{M}(z) = \tilde{d}(z)\tilde{d}(1/z)$. (If we have an (anti)symmetric Laurent polynomial d such that $\det M(z) = d(z)d(1/z)$, then we can take $\tilde{d}(z) = d(z)/(g(z)d_h(z))$).*

(4) *Assume $\tilde{A}(z) = \tilde{A}_0 + \sum_{k=1}^N \tilde{A}_k(z^k + z^{-k})$ with $\tilde{A}_N \neq 0$. Parameterizing the (anti)symmetric Laurent polynomials $\tilde{u}_1$ and $\tilde{v}_1$ such that $[S\tilde{u}_1](z)[S\tilde{v}_1](z) = [S\tilde{B}](z)[S\tilde{d}](z)$ and the degrees of $\tilde{u}_1$ and $\tilde{v}_1$ are at most N (see the paragraph after*

the formula (4.5) *about how to parameterize* $\tilde{u}_1$ *and* $\tilde{v}_1$*). Then according to Theorem* 4.1 *there must be a nonzero solution* $\{\tilde{u}_1, \tilde{v}_1\}$ *to the system of linear homogeneous equations derived from*

$$\tilde{B}(1/z)\tilde{u}_1(z) - \tilde{d}(z)\tilde{v}_1(z) \equiv 0 \qquad \mod \quad \tilde{A}(z).$$

By the proof of Theorem 4.1*, we must have* $\tilde{u}_1(1)^2 + \tilde{v}_1(1)^2 \neq 0$*. Multiplying* $\tilde{u}_1$ *and* $\tilde{v}_1$ *by a constant, we can require that the solution* $\{\tilde{u}_1, \tilde{v}_1\}$ *satisfy* $\tilde{u}_1(1)^2 + \tilde{v}_1(1)^2 = \tilde{A}(1)$.

(5) *Define the symmetric filters* $\tilde{u}_2$ *and* $\tilde{v}_2$ *by*

$$\tilde{u}_2(z) := \frac{\tilde{B}(1/z)\tilde{u}_1(z) - \tilde{d}(z)\tilde{v}_1(1/z)}{\tilde{A}(z)} \quad \textit{and}$$
$$\tilde{v}_2(z) := \frac{\tilde{d}(z)\tilde{u}_1(1/z) + \tilde{B}(1/z)\tilde{v}_1(z)}{\tilde{A}(z)}.$$

(6) *By Lemma* 4.4*, write* $g(z) = q_1(z)q_1(1/z) + q_2(z)q_2(1/z)$ *for some (anti)symmetric Laurent polynomials* q_1 *and* q_2 *such that* $[Sq_1](z)/[Sq_2](z) = [S\tilde{v}_1](z)/[S\tilde{u}_1](z)$*. (In most cases,* $g = 1$ *and we can simply choose* $q_1 = 1$ *and* $q_2 = 0$*.)*

(7) *Obtain the (anti)symmetric Laurent polynomials (or symmetric FIR filters)* u_1*,* u_2*,* v_1*,* v_2 *by*

$$U(z) := \begin{bmatrix} u_1(z) & v_1(z) \\ u_2(z) & v_2(z) \end{bmatrix} = \begin{bmatrix} d_h(z) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{u}_1(z) & \tilde{v}_1(z) \\ \tilde{u}_2(z) & \tilde{v}_2(z) \end{bmatrix} \begin{bmatrix} q_1(z) & -q_2(1/z) \\ q_2(z) & q_1(1/z) \end{bmatrix}.$$

Then $U(z)U(1/z)^T = M(z)$ *and* $[Su_1](z)[Sv_2](z) = [Su_2](z)[Sv_1](z)$.

It is not necessary to check all the conditions in Theorem 2.3 in advance. If at some step one cannot carry out Algorithm 5.1, then the conditions in Theorem 2.3 cannot be satisfied.

Acknowledgments. The authors would like to thank the referees for helpful comments to improve the presentation of this paper and for suggesting the reference [16].

REFERENCES

[1] C. K. Chui and W. He, *Compactly supported tight frame associated with refinable functions*, Appl. Comput. Harmon. Anal., 8 (2000), pp. 293–319.
[2] C. K. Chui, W. He, and J. Stöckler, *Compactly supported tight and sibling frames with maximum vanishing moments*, Appl. Comput. Harmon. Anal., 13 (2002), pp. 224–262.
[3] I. Daubechies, *The wavelet transform, time-frequency localization and signal analysis*, IEEE Trans. Inform. Theory, 36 (1990), pp. 961–1005.
[4] I. Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Regional Conf. Ser. in Appl. Math. 61, SIAM, Philadelphia, 1992.
[5] I. Daubechies and B. Han, *Pairs of dual wavelet frames from any two refinable functions*, Contr. Approx., to appear.
[6] I. Daubechies, B. Han, A. Ron, and Z. W. Shen, *Framelets: MRA-based constructions of wavelet frames*, Appl. Comp. Harmon. Anal., 14 (2003), pp. 1–46.
[7] B. Han, *Compactly supported tight wavelet frames and orthonormal wavelets of exponential decay with a general dilation matrix*, J. Comput. Appl. Math., 155 (2003), pp. 43–67.
[8] B. Han, *On dual wavelet tight frames*, Appl. Comput. Harmon. Anal., 4 (1997), pp. 380–413.
[9] B. Han and Q. Mo, *Tight wavelet frames generated by three symmetric B-spline functions with high vanishing moments*, Proc. Amer. Math. Soc., 132 (2004), pp. 77–86.
[10] D. P. Hardin, T. A. Hogan, and Q. Y. Sun, *The Matrix-Valued Riesz Lemma and Local Orthonormal Bases in Shift-Invariant Spaces*, preprint.

[11] Q. T. Jiang, *Parameterizations of masks for tight affine frames with two symmetric/antisymmetric generators*, Adv. Comput. Math., 18 (2003), pp. 247–268.
[12] A. Petukhov, *Explicit construction of framelets*, Appl. Comput. Harmon. Anal., 11 (2001), pp. 313–327.
[13] A. Petukhov, *Symmetric Framelets*, preprint, 2001.
[14] A. Ron and Z. W. Shen, *Affine systems in $L_2(\mathbb{R}^d)$: The analysis of the analysis operator*, J. Funct. Anal., 148 (1997), pp. 408–447.
[15] I. Selesnick, *Smooth wavelet tight frames with zero moments*, Appl. Comput. Harmon. Anal., 10 (2001), pp. 163–181.
[16] Y. Q. Shi and N. K. Bose, *Nonnegativity constrained spectral factorization for image reconstruction from autocorrelation data*, in Proceedings of International Conference on Acoustics, Speech, and Signal Processing, 1988, pp. 1766–1769.
[17] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice-Hall, Englewood Cliffs, NJ, 1993.
[18] M. Vetterli and J. Kovacévic, *Wavelets and Subband Coding*, Prentice-Hall Signal Processing Series 48, Prentice-Hall, Englewood Cliffs, NJ, 1995.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 1, pp. 125–153

INEXACT KRYLOV SUBSPACE METHODS FOR LINEAR SYSTEMS*

JASPER VAN DEN ESHOF† AND GERARD L. G. SLEIJPEN‡

Abstract. There is a class of linear problems for which the computation of the matrix-vector product is very expensive since a time consuming method is necessary to approximate it with some prescribed relative precision. In this paper we investigate the impact of approximately computed matrix-vector products on the convergence and attainable accuracy of several Krylov subspace solvers. We will argue that the sensitivity towards perturbations is mainly determined by the underlying way the Krylov subspace is constructed and does not depend on the optimality properties of the particular method. The obtained insight is used to tune the precision of the matrix-vector product in every iteration step in such a way that an overall efficient process is obtained. Our analysis confirms the empirically found relaxation strategy of Bouras and Frayssé for the GMRES method proposed in [*A Relaxation Strategy for Inexact Matrix-Vector Products for Krylov Methods*, Technical Report TR/PA/00/15, CERFACS, France, 2000]. Furthermore, we give an improved version of a strategy for the conjugate gradient method of Bouras, Frayssé, and Giraud used in [*A Relaxation Strategy for Inner-Outer Linear Solvers in Domain Decomposition Methods*, Technical Report TR/PA/00/17, CERFACS, France, 2000].

Key words. Krylov subspace methods, inexact matrix-vector product, approximate matrix-vector product, Richardson iteration, Chebyshev iteration, GMRES, FOM, CG, Orthores, residual gap

AMS subject classifications. 65F10

DOI. 10.1137/S0895479802403459

1. Introduction.

There is a class of linear problems where the coefficient matrix cannot be stored explicitly in computer memory but where the matrix-vector products can be computed relatively cheaply using an approximation technique. For this type of problem, direct methods are not attractive. Krylov subspace methods for solving linear systems of equations require, in every iteration step, basic linear algebra operations, like adding vectors and doing inner products, and, usually, one or two matrix-vector products. This makes this class of solution methods very attractive for the mentioned class of problems since we can very easily replace the matrix-vector product in a particular Krylov subspace method with some approximation.

It is obvious that the accurate computation of the matrix-vector product can be quite time consuming if done to high precision. On the other hand, the accuracy of the matrix-vector product has an influence on the Krylov subspace method used for solving the linear system. In this paper we investigate the impact of approximately computed, or inexact, matrix-vector products on the convergence and attainable accuracy of various Krylov subspace methods. Our analysis should provide further insight into the *relaxation strategies* for the accuracy of the matrix-vector product as introduced by Bouras and Frayssé [3] and Bouras, Frayssé, and Giraud [4]. For example, for GMRES they propose to compute the matrix-vector product with a precision proportional to the inverse of the norm of the current residual. When the residual

*Received by the editors March 5, 2002; accepted for publication (in revised form) by Z. Strakoš December 10, 2003; published electronically August 27, 2004.
http://www.siam.org/journals/simax/26-1/40345.html

†Department of Mathematics, Heinrich Heine Universität, Universitätsstr. 1, D-40224, Düsseldorf, Germany (eshof@am.uni-duesseldorf.de). The research of the first author was supported by Dutch Scientific Organization (NWO) project 613.002.035.

‡Department of Mathematics, Utrecht University, P.O. Box 80.010, NL-3508 TA Utrecht, The Netherlands (sleijpen@math.uu.nl).

decreases, the demands on the quality of the computed matrix-vector product are relaxed, which explains the term relaxation. Various researchers have reported that this strategy works remarkably well for practical problems.

The, perhaps, counterintuitive phenomenon that an accurate matrix-vector product is needed in the beginning of the iterative process, instead of at the final iterations has also been observed and analyzed for the Lanczos method for the eigenvalue problem [13]. We also like to refer to independent work of Simoncini and Szyld presented in [25]. This work later resulted in the paper [26] and some comments on the differences with the work described here can be found at the end of this paper.

In this paper we focus on the impact of perturbations on the matrix-vector product in various Krylov subspace solvers. This problem is related to rounding error analysis of Krylov subspace methods since in the latter case an inexact matrix-vector product is one source of errors. In our analysis we will use an approved method from this area: we try to bound the norm of the *residual gap* and separately analyze the behavior of the *computed residuals* (although this is possible only in a few special cases). The usual way for bounding the gap is based on an inspection of the recurrences, e.g., [27, 15, 20, 19, 2]. Our approach differs from the analysis in these papers in the sense that the analysis here is based on exploiting properties of the upper Hessenberg matrices that arise in the matrix formulation of the Krylov subspace method. Where possible we point out the differences with techniques used in literature and discuss implications for rounding error analysis.

Another related problem is when a variable preconditioner is used in the Krylov subspace method. See [10, 24, 31, 9, 12] for some results and the discussion throughout this paper.

The outline of this paper is as follows. In sections 2 and 3 we set up the framework that we need in the rest of this paper. We give an expression for the residual gap for a general Krylov subspace method in section 3. This general expression is exploited in the remainder of this paper, starting with Richardson iteration in section 4 and Chebyshev iteration in section 5. The conjugate gradient (CG) method is the subject of section 6. Inexact GMRES and FOM for general matrices are treated in section 7 and we conclude with some numerical experiments in section 8.

2. Krylov subspace methods. This paper is concerned with the approximate solution of the $n \times n$ linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad \text{with} \quad \|\mathbf{b}\|_2 = 1. \tag{2.1}$$

In this section we summarize some properties (in terms of matrix formulations) of the class of iterative linear system solvers called *Krylov subspace methods*.

Before we continue we have to define some notation. The vector e_k denotes the kth standard basis vector, i.e., $(e_k)_j = 0$ for all $j \neq k$ and $(e_k)_k = 1$. Furthermore, $\vec{1}$ is the vector with all components one and, similarly, $\vec{0}$ is the vector with all components zero. The dimension of these vectors should be apparent from the context. We warn the reader for some unconventional notation: if we apply a matrix with k columns to an ℓ-vector with $\ell \leq k$, then we assume the vector to be expanded with zeros if necessary (we do the same with other operations and equalities). Finally, we use bold capital letters to denote matrices with n rows and use small bold capitals to denote the columns of these matrices where the subscript indicates the column number (starting with 0), so, for example, $\mathbf{v}_0 = \mathbf{V}e_1$. The zero vector of length n is denoted by $\mathbf{0}$.

The notion of a *Krylov subspace* plays an important role in the analysis and derivation of a large class of iterative methods for solving (2.1). The Krylov subspace

of order k (generated by the matrix $\mathbf{A}$ and the vector $\mathbf{b}$) is defined as

$$\mathcal{K}_k \equiv \mathcal{K}_k(\mathbf{A}, \mathbf{b}) \equiv \text{span}\{\mathbf{b}, \mathbf{A}\mathbf{b}, \ldots, \mathbf{A}^{k-1}\mathbf{b}\}. \tag{2.2}$$

In this paper we concentrate on iterative solution methods for which the iterate in step j, $\mathbf{x}_j$, and its corresponding residual $\mathbf{r}_j = \mathbf{b} - \mathbf{A}\mathbf{x}_j$, respectively, belong to the spaces $\mathcal{K}_j$ and $\mathcal{K}_{j+1}$. Iterative solution methods with this property are called Krylov subspace methods.[1] We, furthermore, assume for all $j \leq k$ that the residuals provide a sequence that after k steps of the subspace method can be summarized by the following matrix relation:

$$\mathbf{A}\mathbf{R}_k = \mathbf{R}_{k+1}\underline{S}_k, \quad \text{with} \quad \mathbf{R}_k e_1 = \mathbf{b}, \quad \vec{1}^*\underline{S}_k = \vec{0}^*. \tag{2.3}$$

Here, the matrix $\mathbf{R}_k$ is an n by k matrix with as jth column $\mathbf{r}_{j-1}$, and $\underline{S}_k$ is a $k+1$ by k upper Hessenberg matrix. The last condition in (2.3) for the Hessenberg matrix is a necessary and sufficient condition for the vector $\mathbf{r}_j$ to be a residual that corresponds to some approximate solution from the space $\mathcal{K}_j$; see [18, section 4.4]. Indeed, if S_j denotes the matrix $\underline{S}_j$ from which the last row is dropped, then, if S_j is invertible, we have with $\beta \equiv e_{j+1}^*\underline{S}_j e_j$,

$$\vec{0}^* = \vec{1}^*\underline{S}_j = \vec{1}^* S_j + \beta e_j^* \quad \Rightarrow \quad \beta e_j^* S_j^{-1} = -\vec{1}^*$$

and

$$\underline{S}_j S_j^{-1} e_1 = \begin{bmatrix} S_j \\ \beta e_j^* \end{bmatrix} S_j^{-1} e_1 = e_1 - e_{j+1}. \tag{2.4}$$

Now, if we let

$$\mathbf{x}_j \equiv \mathbf{R}_j(S_j^{-1} e_1), \tag{2.5}$$

then we get, using (2.3) and (2.4), that

$$\begin{aligned} \mathbf{b} - \mathbf{A}\mathbf{x}_j &= \mathbf{b} - \mathbf{A}\mathbf{R}_j(S_j^{-1}e_1) = \mathbf{b} - \mathbf{R}_{j+1}(\underline{S}_j S_j^{-1} e_1) \\ &= \mathbf{b} - \mathbf{R}_{j+1}(e_1 - e_{j+1}) = \mathbf{b} - (\mathbf{r}_0 - \mathbf{r}_j) = \mathbf{r}_j. \end{aligned}$$

This shows that $\mathbf{r}_j = \mathbf{b} - \mathbf{A}\mathbf{x}_j$ if $\mathbf{x}_j$ is as in (2.5). Hence, for this choice we can say that the iterate $\mathbf{x}_j$ is *consistent* with the residual vector $\mathbf{r}_j$.

Moreover, we can get a recursion for the iterates $\mathbf{x}_j$ by substituting $\mathbf{R}_k = \mathbf{b}\vec{1}^* - \mathbf{A}\mathbf{X}_k$ in (2.3). This shows that

$$-\mathbf{R}_k = \mathbf{X}_{k+1}\underline{S}_k, \quad \mathbf{X}_k e_1 = \mathbf{0}. \tag{2.6}$$

Some Krylov subspace methods use the recursions in (2.3) or (2.6) explicitly in their implementation. An example is the Chebyshev method where the iterates are computed with the, in this case, three-term relation in (2.6); see also section 5.

It is common to view Krylov subspace methods as polynomial based iteration methods where the residuals are characterized as matrix polynomials in $\mathbf{A}$ that act on the vector $\mathbf{b}$; see, e.g., [6]. This viewpoint plays an important role in the convergence analysis of a large number of Krylov subspace methods. The property of $\underline{S}_k$ that the

[1] Notice that this characterization does not include the Bi-CGSTAB method, for example.

columns sum up to zero, is equivalent to the fact that the residual polynomials have the interpolatory constraint that they are one in zero. We will, however, not use this polynomial interpretation and will mostly consider the matrix formulation and exploit algebraic properties of the matrix $\underline{S}_k$.

We conclude this section with a useful property of the Hessenberg matrix $\underline{S}_k$ that we will frequently use in the remainder of this paper.

LEMMA 2.1. *If the matrix S_j is invertible for $j \leq k$, then the LU-decomposition of S_k and the one of $\underline{S}_k$ exists. Furthermore,*

$$S_k = J_k U_k \quad \text{and} \quad \underline{S}_k = \underline{J}_k U_k, \tag{2.7}$$

where $\underline{J}_k$ is lower bidiagonal with $(\underline{J}_k)_{j,j} = 1$ and $(\underline{J}_k)_{j+1,j} = -1$ and U_k is upper triangular with $(U_k)_{i,j} = \sum_{l=1}^{i} (\underline{S}_k)_{l,j}$ for $i \leq j$.

Proof. The existence of the LU-decomposition of S_k follows from the fact that each principal submatrix of S_k is nonsingular; see, for instance, [11, Theorem 3.2.1]. The matrix J_k^{-1} is lower triangular with all components one. Therefore, it follows that $J_k^{-1} S_k = U_k$. This proves the first equality in (2.7). The second equality follows by checking that

$$\underline{J}_k U_k = (J_k - e_{k+1} e_k^*) U_k = S_k - e_{k+1} e_k^* U_k = \underline{S}_k. \qquad \square$$

2.1. Derivation from Krylov decompositions. For theoretical purposes and future convenience, we summarize in this section some facts about a so-called *Krylov decomposition* given by

$$\mathbf{A}\mathbf{C}_k = \mathbf{C}_{k+1} \underline{T}_k, \quad \mathbf{C}_k e_1 = \mathbf{b}, \tag{2.8}$$

where $\mathbf{C}_k$ is an n by k matrix and $\underline{T}_k$ is a $k+1$ by k upper Hessenberg matrix. The column space of $\mathbf{C}_k$ is a subspace of the Krylov space $\mathcal{K}_k$ but the columns, $\mathbf{c}_j$, are not necessarily residuals corresponding to approximations from $\mathcal{K}_j$. However, from this relation different residual sequences (2.3) can be derived depending on the required properties for the $\mathbf{r}_j$. In order to continue our discussion, we assume that $\underline{T}_k$ has full rank, and we define the $k+1$-vector $\vec{\gamma}_k$ as the vector such that $\vec{\gamma}_k^* \underline{T}_k = \vec{0}^*$ and $\vec{\gamma}_k^* = (1, \gamma_1, \ldots, \gamma_k)^*$. Notice that, due to the Hessenberg structure of $\underline{T}_k$, the elements γ_j can be computed using a simple and efficient recursion.

A simple way to derive a residual sequence is to put $\Gamma_k \equiv \mathrm{diag}(\vec{\gamma}_{k-1})$; then we see that the matrices

$$\underline{S}_k \equiv \Gamma_{k+1} \underline{T}_k \Gamma_k^{-1} \quad \text{and} \quad \mathbf{R}_k \equiv \mathbf{C}_k \Gamma_k^{-1} \tag{2.9}$$

satisfy (2.3) (with, indeed, $\vec{1}^* \underline{S}_k = \vec{0}^*$). In this case the residual $\mathbf{r}_j$ is a multiple of the vector $\mathbf{c}_j$. In terms of the polynomial interpretation of Krylov subspace methods, this construction of the residual sequence can be viewed as obtaining the residual polynomials by scaling the polynomials, generated by the coefficients in $\underline{T}_k$, such that they are one in zero. Furthermore, if T_j is invertible, then we have for the residual

$$\mathbf{r}_j = \mathbf{c}_j / \gamma_j = \mathbf{C}_{j+1} (I - \underline{T}_j T_j^{-1}) e_1 = \mathbf{b} - \mathbf{A}\mathbf{C}_j T_j^{-1} e_1, \tag{2.10}$$

where we have used (2.8) and the first statement of the following lemma. (For ease of future reference, we formulate the lemma slightly more general than needed here.)

LEMMA 2.2. *Let $j \leq k$. Then,*

$$e_1 - \underline{T}_j(T_j^{-1}e_1) = \frac{e_{j+1}}{\gamma_j} \quad \textit{and} \quad e_1 - \underline{T}_j(\underline{T}_j^{\dagger}e_1) = \frac{\vec{\gamma}_j}{\|\vec{\gamma}_j\|_2^2}, \tag{2.11}$$

where $\underline{T}_j^{\dagger}$ denotes the generalized inverse of $\underline{T}_j$ [11, section 5.5.4] *and where, for the first expression, T_j is assumed to be invertible.*

Proof. The first expression follows from a combination of $e_1 - \underline{T}_j(T_j^{-1}e_1) = e_1 - \Gamma_{j+1}^{-1}\underline{S}_j S_j^{-1}\Gamma_j e_1$ and (2.4). For the second expression we notice that $I - \underline{T}_j\underline{T}_j^{\dagger}$ is the orthogonal projection on $\mathrm{Ker}(\underline{T}_j{}^*) = \mathrm{span}(\vec{\gamma}_j)$, we have that $I - \underline{T}_j\underline{T}_j^{\dagger} = \|\vec{\gamma}_j\|_2^{-2}\,\vec{\gamma}_j\,\vec{\gamma}_j^*$. This leads to the first expression in (2.11). ∎

The lemma also leads to an expression for residuals from an alternative construction:

$$\mathbf{r}_j = \mathbf{b} - \mathbf{A}\mathbf{C}_j\underline{T}_j^{\dagger}e_1 = \mathbf{C}_{j+1}(I - \underline{T}_j\underline{T}_j^{\dagger})e_1 = \frac{1}{\|\vec{\gamma}_j\|_2^2}\mathbf{C}_{j+1}\vec{\gamma}_j. \tag{2.12}$$

If we define

$$\Upsilon_k \equiv [\vec{\gamma}_0, \ldots, \vec{\gamma}_{k-1}], \quad \Theta_k \equiv \mathrm{diag}(\|\vec{\gamma}_0\|_2, \ldots, \|\vec{\gamma}_{k-1}\|_2),$$

then we get

$$\underline{S}_k \equiv (\Upsilon_{k+1}\Theta_{k+1}^{-2})^{-1}\underline{T}_k(\Upsilon_k\Theta_k^{-2}) \quad \text{and} \quad \mathbf{R}_k \equiv \mathbf{C}_k(\Upsilon_k\Theta_k^{-2}). \tag{2.13}$$

It can be easily checked that $\vec{1}^*(\Upsilon_{k+1}\Theta_{k+1}^{-2})^{-1} = \vec{\gamma}_k^*$ and therefore $\vec{1}^*\underline{S}_k = \vec{0}^*$ and also the Hessenberg form is preserved. It should be noted that the matrix $(\Upsilon_{k+1}\Theta_{k+1}^{-2})^{-1}$ can be decomposed into simple factors since $\Upsilon_{k+1} = \Gamma_{k+1}J_{k+1}^{-1}$. These latter observations are related to the well-known fact (see, e.g., [6, section 2.5]) that *minimal residual* polynomials, or *Kernel* polynomials, can be generated efficiently using coupled recurrences.

3. Inexact Krylov subspace methods. In the previous section we collected some general properties of Krylov subspace methods. There is a class of applications for which it is very costly to compute the matrix-vector product to high precision. The original motivation for the research in this paper was a linear system that occurs in simulations in quantum chromodynamics (QCD) [8]. In this area the so-called *overlap formulation* has initiated a lot of research in solving linear systems of the form

$$(r\mathbf{\Gamma_5} + \mathrm{sign}(\mathbf{Q}))\mathbf{x} = \mathbf{b}, \quad \|\mathbf{b}\| = 1 \quad (r \geq 1), \tag{3.1}$$

where $\mathbf{Q}$ and $\mathbf{\Gamma_5}$ are sparse Hermitian indefinite matrices. The matrix $\mathrm{sign}(\mathbf{Q})$ is the so-called *matrix sign function*; see, e.g., [11, p. 372]. This matrix is dense and is known only implicitly since we are given only the action of the matrices $\mathbf{Q}$ and $\mathbf{\Gamma_5}$ to vectors. Realistic simulations require in the order of one to ten million unknowns. Usually, (3.1) is solved with a standard Krylov subspace method for linear systems, for example the CG method (since this matrix is Hermitian). In every step some vector iteration method is required to compute the product of $\mathrm{sign}(\mathbf{Q})$ and a vector. The usual approach is to construct some polynomial approximation for the sign function, for example with a Lanczos approximation. For an overview and comparison of methods used in this context we refer to [30].

In this paper we consider the general problem of solving (2.1) where we assume that we are given, for every scalar η and vector y, some approximation function $\mathcal{M}_\eta : \mathbb{C}^n \to \mathbb{C}^n$ with the property that

$$\mathcal{M}_\eta(\mathbf{y}) = \mathbf{A}\mathbf{y} + \mathbf{g} \quad \text{with} \quad \|\mathbf{g}\|_2 \leq \eta \|\mathbf{A}\|_2 \, \|\mathbf{y}\|_2. \tag{3.2}$$

It is, furthermore, assumed that the smaller η is chosen, the more time consuming this approximation becomes to construct.

In the iterative methods that we discuss, it is necessary in step j to compute the product of the matrix $\mathbf{A}$ with some vector, say $\mathbf{y}$. If the matrix-vector products are replaced with approximations computed with the function $\mathcal{M}_\eta$, then we will refer to the resulting method as an *inexact* Krylov subspace method. This can also be viewed as a Krylov subspace method where a perturbation $\mathbf{g}_{j-1}$ is added to the exact matrix-vector product in step j where $\mathbf{g}_{j-1}$ is such that $\|\mathbf{g}_{j-1}\|_2 \leq \eta_{j-1} \|\mathbf{A}\|_2 \, \|\mathbf{y}\|_2$.

Due to the existence of the errors, $\mathbf{g}_{j-1}$, the space spanned by the residuals computed in the iterative method, is, in general, not a Krylov subspace generated by $\mathbf{A}$ anymore. This has two consequences: the convergence behavior is altered, and the maximally attainable accuracy of the iterative method is limited. The central question in this paper is how large the perturbations can be if one is interested in a solution $\mathbf{x}_k$ such that $\|\mathbf{b} - \mathbf{A}\mathbf{x}_k\|_2 = \mathcal{O}(\varepsilon)$ without altering the convergence behavior too much, or equivalently, how to pick η_{j-1} in step j.

3.1. Relaxation strategies. In [3], Bouras and Frayssé showed numerical experiments for GMRES with a relative precision η_j in step $j+1$ given by

$$\eta_j = \max\left\{ \frac{\varepsilon}{\|\mathbf{b} - \mathbf{A}\mathbf{x}_j\|_2}, \varepsilon \right\}. \tag{3.3}$$

For an impressive list of numerical experiments, they observed that with (3.3) the GMRES method converged roughly as fast as the unperturbed version, despite the sometimes large perturbations. Furthermore, the norm of the true residual ($\|\mathbf{b} - \mathbf{A}\mathbf{x}_j\|_2$) seems to stagnate around a value of $\mathcal{O}(\varepsilon)$. Obviously, such a strategy can result in large savings in practical applications. The true residual is unfortunately, in general, not known, since this would require an exact matrix-vector product. The approximate residual, as computed in the inexact Krylov subspace method (cf. section 3.2), can serve as an alternative. Another interesting property of this choice for η_j is that it requires very accurate matrix-vector products in the beginning of the process, and the precision is relaxed as soon as the method starts to converge; that is, the residuals become small. This justifies the term *relaxation strategy* as introduced in [3]. We conclude with the remark that this condition was derived empirically in [3] based on the experience of the authors with a large number of experiments and no insight or analysis is given to explain this remarkable observation.

3.2. The analysis of inexact Krylov subspace methods. In the remainder of this paper we will see that, for the methods that we consider, the approximate residuals, $\mathbf{r}_j$, computed in the inexact Krylov subspace method now satisfy the perturbed relation

$$\mathbf{A}\mathbf{R}_k + \mathbf{F}_k = \mathbf{R}_{k+1}\underline{S}_k, \quad \text{with} \quad \mathbf{R}_k e_1 = \mathbf{b}, \quad \vec{1}^* \underline{S}_k = \vec{0}^*. \tag{3.4}$$

The columns of the matrix $\mathbf{F}_k$ are a function of the errors in the matrix-vector products. Furthermore, $\mathbf{x}_j$ still satisfies (2.5) (or equivalently (2.6)) because of the assumption of exact arithmetic. For the moment we assume that these relations hold

but we stress that their validity must be checked for every inexact Krylov subspace method which is obtained by replacing in a particular method the exact matrix-vector product with some approximation.

As a consequence of the perturbation term $\mathbf{F}_k$, the vector $\mathbf{r}_k$ is usually not a residual anymore for the approximate solution $\mathbf{x}_k$. Therefore, we will refer to the vector $\mathbf{r}_k$ as the *computed residual* in contrast to the *true residual* defined by $\mathbf{b}-\mathbf{A}\mathbf{x}_k$. In the analysis of inexact Krylov methods, the true residuals are the quantities of interest and we have

$$\|\mathbf{b}-\mathbf{A}\mathbf{x}_k\|_2 \le \|\mathbf{r}_k\|_2 + \|\mathbf{r}_k-(\mathbf{b}-\mathbf{A}\mathbf{x}_k)\|_2. \tag{3.5}$$

This inequality forms the basis of our analysis. If the computed residuals, for sufficiently large k, become small compared to the residual gap, then it follows from (3.5) that the stagnation level of the inexact Krylov subspace method is determined by the *residual gap*, the difference between the computed residual and the true residual. Furthermore, in the early iterations the norm of the computed residuals is large compared to the size of the residual gap. This shows that the initial convergence of the true residuals is determined by the residuals computed in the inexact Krylov subspace method.

In the coming sections we will analyze the effect of inexact matrix-vector products and, in particular, relaxation strategies as in (3.3) on different Krylov subspace methods by writing the residual relation into the form (3.4) and by bounding the residual gap. If it is additionally shown that the computed residuals in the end become sufficiently small, then the residual gap will ultimately determine the attainable accuracy. The convergence of the computed residuals is a difficult topic that we can only fully analyze in some special cases. It should be noticed that for the applications that we have in mind, the norm of the computed residuals can be efficiently monitored, while for the true residual or size of the residual gap, it is necessary to compute an accurate matrix-vector product which is not feasible. It turns out that, under our assumptions, a general expression can be given for the residual gap. We give this expression in section 3.3 and exploit it in the remainder of this paper.

For the analysis in this paper, we assume the use of exact arithmetic operations. Here, we are interested in the effect of errors in the matrix-vector multiplication, but it is also a reasonable assumption, considering that, in general, the "error" in the matrix-vector product is much larger than machine precision, as in the QCD example (3.1) mentioned in the beginning of section 3, where the error in the matrix-vector product is an error resulting from the truncation of an approximation process for the matrix sign function times a vector.

3.3. A general expression for the residual gap. The goal is to get an expression for the residual gap. Assuming that $\mathbf{x}_k$ is of the form (2.6) and the computed residuals satisfy (3.4), then we find, using again (2.4), the following expression:

$$\mathbf{r}_k-(\mathbf{b}-\mathbf{A}\mathbf{x}_k) = \mathbf{r}_k-\mathbf{r}_0+\mathbf{A}\mathbf{R}_k S_k^{-1}e_1 = -\mathbf{F}_k S_k^{-1}e_1 = -\sum_{j=1}^{k}\mathbf{f}_{j-1}e_j^* S_k^{-1}e_1. \tag{3.6}$$

This shows that the expression for the gap is a linear combination of the columns of $\mathbf{F}_k$, i.e., the vectors $\mathbf{f}_{j-1}$. The coefficients $-(e_j^* S_k^{-1}e_1)$ somehow determine the propagation of the perturbations through the recurrences. Our approach for bounding the gap is based on using properties of the matrix $\underline{S}_k$. We will do this for various

Krylov subspace methods in the remainder of this paper. Therefore, the following lemma is convenient and will frequently be used.

LEMMA 3.1. *Let $\underline{T}_k$ be upper Hessenberg and of full rank. For $j \leq k$, we have*

$$|e_j^* \underline{T}_k^\dagger e_1| \leq \|\underline{T}_k^\dagger\|_2 \frac{1}{\|\vec{\gamma}_{j-1}\|_2}, \quad |e_j^* T_k^{-1} e_1| \leq \|\underline{T}_k^\dagger\|_2 \left(\frac{1}{\|\vec{\gamma}_{j-1}\|_2} + \frac{1}{|\gamma_k|} \right). \tag{3.7}$$

Proof. To prove (3.7), we observe that $\underline{T}_k^\dagger \underline{T}_k$ is the identity on k-vectors if $\underline{T}_k$ is of rank k. Since $e_j^* \vec{y}_{j-1} = 0$ for any $j-1$-vector $\vec{y}_{j-1}$ we have that

$$\begin{aligned} e_j^* \underline{T}_k^\dagger e_1 &= e_j^* \underline{T}_k^\dagger (e_1 - \underline{T}_k \vec{y}_{j-1}) \quad \text{and} \\ e_j^* T_k^{-1} e_1 &= e_j^* \underline{T}_k^\dagger (e_1 - \underline{T}_k \vec{y}_{j-1}) + e_j^* \underline{T}_k^\dagger (\underline{T}_k (T_k^{-1} e_1) - e_1). \end{aligned}$$

With $\vec{y}_{j-1} = \underline{T}_{j-1}^\dagger e_1$ and $\vec{y}_{j-1} = T_{j-1}^{-1} e_1$, a combination with (2.11) leads to

$$e_j^* \underline{T}_k^\dagger e_1 = e_j^* \underline{T}_k^\dagger \frac{\vec{\gamma}_{j-1}}{\|\vec{\gamma}_{j-1}\|_2^2} = e_j^* \underline{T}_k^\dagger \frac{e_j}{\gamma_{j-1}} \text{ and } e_j^* T_k^{-1} e_1 = e_j^* \underline{T}_k^\dagger e_1 - e_j^* \underline{T}_k^\dagger \frac{e_{k+1}}{\gamma_k},$$

and (3.7) easily follows. ∎

We expressed our estimates in terms of the smallest singular value of $\underline{T}_k$. This value depends monotonically (decreasing) on k, and $\|T_m^{-1}\|_2 \geq \|\underline{T}_k{}^\dagger\|_2$ if $m > k$. The smallest singular value of T_k does not have this attractive property: even if T_m is well-conditioned, there may be a $k < m$ for which T_k is singular or nearly singular.

4. Inexact Richardson iteration. One of the simplest iterative methods for linear systems is *Richardson iteration*, e.g., [16]. This method allows a straightforward analysis, however, it already demonstrates some important aspects of our analysis. Therefore, Richardson iteration is useful as a starting point. With a perturbed matrix-vector product, this method is described by the following recurrences for $j = 1, \ldots, k$ (with $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{r}_0 = \mathbf{b}$):

$$\mathbf{r}_j = \mathbf{r}_{j-1} - \alpha(\mathbf{A}\mathbf{r}_{j-1} + \mathbf{g}_{j-1}), \tag{4.1}$$

$$\mathbf{x}_j = \mathbf{x}_{j-1} + \alpha \mathbf{r}_{j-1}, \tag{4.2}$$

and $\|\mathbf{g}_j\| \leq \eta_j \|\mathbf{A}\|_2 \|\mathbf{r}_j\|_2$. For simplicity we restrict our attention to symmetric positive definite matrices $\mathbf{A}$ with an optimal choice for α:

$$\alpha \equiv \frac{2}{\lambda_{\min} + \lambda_{\max}}, \tag{4.3}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are, respectively, the smallest and largest eigenvalue of $\mathbf{A}$.

For this method it is clear that after k steps of the method, the iterates satisfy (2.6) and the residuals satisfy (3.4) with $\mathbf{F}_k = \mathbf{G}_k$ and $\underline{S}_k = \underline{J}_k U_k$ with $U_k = \alpha^{-1} I$. Therefore, we can exploit (3.6) and, using $e_j^* S_k^{-1} e_1 = \alpha$, we get the following bound on the norm of the residual gap:

$$\|\mathbf{r}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_k)\|_2 = \left\| \sum_{j=1}^{k} \mathbf{f}_{j-1} \alpha \right\|_2 \leq \alpha \|\mathbf{A}\|_2 \sum_{j=0}^{k-1} \eta_j \|\mathbf{r}_j\|_2.$$

Recall that we are only interested in an approximate solution $\mathbf{x}_k$ with $\|\mathbf{b} - \mathbf{A}\mathbf{x}_k\|_2 = \mathcal{O}(\varepsilon)$. This suggests to pick $\eta_j = \varepsilon / \|\mathbf{r}_j\|_2$ and for this choice we get, using (4.3),

$$\|\mathbf{r}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_k)\|_2 \leq \varepsilon k \alpha \|\mathbf{A}\|_2 = \varepsilon 2k \frac{\mathcal{C}(\mathbf{A})}{\mathcal{C}(\mathbf{A}) + 1} < \varepsilon 2k,$$

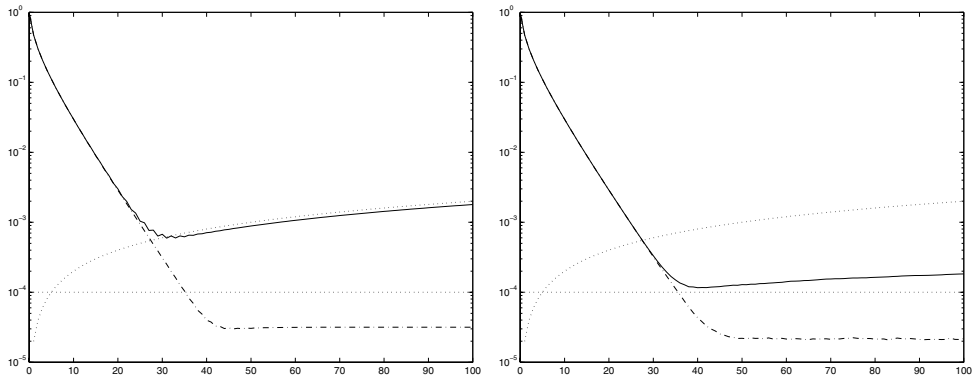


FIG. 4.1. *Richardson iteration with* $\eta_j = 10^{-5}/\|\mathbf{r}_j\|_2$, *true residuals* (—), *norm computed residual* (- ·), *and the quantities* $10^{-5}\mathcal{C}(\mathbf{A})$, $2j10^{-5}$ *(both dotted) as a function of* j. *The matrix* $\mathbf{A}$ *has dimension* 1000 *and* $\mathcal{C}(\mathbf{A}) = 10$. *Left: Errors have all components equal. Right: Random errors.*

where $\mathcal{C}(\mathbf{A}) \equiv \|\mathbf{A}\|_2 \, \|\mathbf{A}^{-1}\|_2$. We stress that the residual gap for this simple iteration method can be obtained by comparing the recursions for $\mathbf{r}_j$ and $\mathbf{b} - \mathbf{A}\mathbf{x}_j$ directly. We have used here a slightly more involved approach to demonstrate the use of our general formula (3.6), which becomes more convenient when studying more advanced methods.

It remains to be shown that the computed residuals become sufficiently small. For inexact Richardson iteration we have the following result which even shows that the computed residuals become small at a speed comparable to the exact process.

THEOREM 4.1. *Let* $\overline{\mathbf{r}}_k$ *satisfy* (4.1) *with* $\eta_j = 0$, *and let* $\mathbf{r}_k$ *satisfy* (4.1) *with* $\eta_j = \varepsilon/\|\mathbf{r}_j\|_2$. *Then*

$$\|\mathbf{r}_k - \overline{\mathbf{r}}_k\| \leq \varepsilon \mathcal{C}(\mathbf{A}).$$

Proof. The difference between the two residuals is given by

$$\mathbf{r}_k - \overline{\mathbf{r}}_k = (I - \alpha\mathbf{A})^k \mathbf{b} + \alpha \sum_{j=1}^{k} (I - \alpha\mathbf{A})^{k-j} \mathbf{f}_{j-1} - (I - \alpha\mathbf{A})^k \mathbf{b} = \alpha \sum_{j=1}^{k} (I - \alpha\mathbf{A})^{k-j} \mathbf{f}_{j-1}.$$

For $\eta_j = \varepsilon/\|\mathbf{r}_j\|_2$ we have $\|\mathbf{f}_j\|_2 \leq \eta_j \|\mathbf{A}\|_2 \, \|\mathbf{r}_j\|_2 = \varepsilon \|\mathbf{A}\|_2$; hence

$$\|\mathbf{r}_k - \overline{\mathbf{r}}_k\|_2 \leq |\alpha| \sum_{j=1}^{k} \|(I - \alpha\mathbf{A})\|_2^{k-j} \varepsilon \|\mathbf{A}\|_2 \leq \varepsilon \|\mathbf{A}\|_2 \, \|(\alpha\mathbf{A})^{-1}\|_2 |\alpha| = \varepsilon \mathcal{C}(\mathbf{A}). \qquad \square$$

Since $\overline{\mathbf{r}}_k$ will go to zero for $k \to \infty$, we expect the norm of $\mathbf{r}_k$ ultimately to stagnate at a level below $\varepsilon\mathcal{C}(\mathbf{A})$. This shows that the final residual precision is essentially determined by the residual gap. We give a simple illustration of this in Figure 4.1, where we have simulated inexact matrix-vector multiplications by adding an artificial perturbation to the exact matrix-vector product. We conclude that for Richardson iteration the required precision of the matrix-vector product can be relaxed with a strategy similar to the one proposed for GMRES in (3.3).

4.1. Discussion. One might remark that in practical applications the residual is not computed in an incremental fashion as in (4.1). However, incrementally computed residuals are important for a relaxation strategy to be successful. Furthermore, directly computed residuals are not necessarily more accurate even if using a fixed precision, i.e., $\eta_j = \eta$. In this case a direct computation of the $(k+1)$th residual yields

$$\|\mathbf{r}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_k)\|_2 \leq \eta \|\mathbf{A}\|_2 \, \|\mathbf{x}_k\|_2 = \|(\eta \|\mathbf{A}\|_2 \mathbf{R}_k) S_k^{-1} e_1\|_2,$$

whereas an expression for the recursively computed residual follows from (3.6)

$$\|\mathbf{r}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_k)\|_2 = \|\mathbf{F}_k S_k^{-1} e_1\|_2.$$

Both $\mathbf{F}_k$ and $\eta\|\mathbf{A}\|_2\mathbf{R}_k$ have a $(j+1)$th column with a length smaller than $\eta\|\mathbf{A}\|_2 \, \|\mathbf{r}_j\|_2$. Hence, the difference in the upper bounds is determined by the mutual angle between the columns. In case the residuals change slowly and if the $\mathbf{f}_j$ are random, the recursively computed residual can be more accurate. Numerical experiments confirm this, although the differences are small. Experiments also suggest that in the situation of only finite precision errors an incrementally computed residual is no longer necessarily more accurate than a directly computed residual as is often observed in practice.

5. Inexact Chebyshev iteration. A more advanced method than Richardson iteration is *Chebyshev iteration*, e.g., [11, section 10.1.5], [7, Chapter 7]. It is more advanced than Richardson iteration in the sense that it employs a three-term recurrence for the residuals for faster convergence. For clarity and in order to establish notation, we start with a short derivation of Chebyshev iteration. Again, we assume $\mathbf{A}$ to be symmetric positive definite.

We define $\phi(t) \equiv \alpha t - \beta$ as a function that maps the interval $[\lambda_{\min}, \lambda_{\max}]$ to the interval $[-1, 1]$, so (for example)

$$\alpha \equiv \frac{2}{\lambda_{\max} - \lambda_{\min}}, \quad \beta \equiv \frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}}. \tag{5.1}$$

The main idea behind the Chebyshev method is to construct the residuals $\mathbf{r}_j$ as multiples of the vectors $\mathbf{c}_j = c_j(\phi(\mathbf{A}))\mathbf{b}$, where $c_j(t)$ is the Chebyshev polynomial of degree j; see [7, p. 4] for a definition. An efficient algorithm comes from the three-term recurrence for the Chebyshev polynomials

$$\mathbf{c}_j = 2\phi(\mathbf{A})\mathbf{c}_{j-1} - \mathbf{c}_{j-2}, \quad \text{with} \quad \mathbf{c}_0 = \mathbf{b}, \ \mathbf{c}_1 = \phi(\mathbf{A})\mathbf{b},$$

which reads in matrix formulation for k steps

$$\mathbf{A}\mathbf{C}_k = \mathbf{C}_k \underline{T}_k \quad \text{with} \quad \underline{T}_k \equiv \begin{bmatrix} \frac{\beta}{\alpha} & \frac{1}{2\alpha} & & \\ \frac{1}{\alpha} & \frac{\beta}{\alpha} & \frac{1}{2\alpha} & \\ & \frac{1}{2\alpha} & \ddots & \ddots \\ & & \ddots & \ddots \\ & & & \frac{1}{2\alpha} \end{bmatrix}. \tag{5.2}$$

Equations (2.3) and (2.9) now give a three-term recurrence for the residuals with $\gamma_j = c_j(\phi(0))$. A recursion for the approximate solutions $\mathbf{x}_j$ is given by (2.6). For

convenience of the reader, we give the resulting recurrence relations: for $j = 2, \ldots, k$, we have

$$\mathbf{r}_j = 2\alpha\frac{\gamma_{j-1}}{\gamma_j}(\mathbf{A}\mathbf{r}_{j-1} + \mathbf{g}_{j-1}) - 2\beta\frac{\gamma_{j-1}}{\gamma_j}\mathbf{r}_{j-1} - \frac{\gamma_{j-2}}{\gamma_j}\mathbf{r}_{j-2}, \tag{5.3}$$

$$\mathbf{x}_j = -2\alpha\frac{\gamma_{j-1}}{\gamma_j}\mathbf{r}_{j-1} - 2\beta\frac{\gamma_{j-1}}{\gamma_j}\mathbf{x}_{j-1} - \frac{\gamma_{j-2}}{\gamma_j}\mathbf{x}_{j-2}, \tag{5.4}$$

with $\mathbf{r}_0 = \mathbf{b}$, $\mathbf{r}_1 = \alpha\frac{\gamma_0}{\gamma_1}(\mathbf{A}\mathbf{r}_0 + \mathbf{g}_0) - \beta\frac{\gamma_0}{\gamma_1}\mathbf{r}_0$, $\mathbf{x}_0 = \mathbf{0}$, and $\mathbf{x}_1 = -\alpha\frac{\gamma_0}{\gamma_1}\mathbf{r}_0$. In this recursion we have already used an inexact version of the matrix-vector product in (5.3). It easily follows that the computed residuals in the inexact Chebyshev method satisfy (3.4) with $\mathbf{F}_k = \mathbf{G}_k$ and therefore $\|\mathbf{f}_j\|_2 \leq \eta_j\|\mathbf{A}\|_2\|\mathbf{r}_j\|_2$. In order to bound the residual gap with (3.6), we have to bound $e_j^* S_k^{-1} e_1$; this is accomplished in the following lemma.

LEMMA 5.1. *Let T_k be as in* (5.2)*, and let α and β be as* (5.1)*. Then*

$$|e_j^* S_k^{-1} e_1| = |e_j^* T_k^{-1} e_j| \leq \frac{2\alpha}{\sqrt{\beta^2 - 1}} = \frac{2}{\sqrt{\lambda_{\max}\lambda_{\min}}} = 2\frac{\sqrt{\mathcal{C}(\mathbf{A})}}{\|\mathbf{A}\|_2}. \tag{5.5}$$

Proof. Using (2.4) we see that

$$e_j^* S_k^{-1} e_1 = e_j^* S_k^{-1}(e_1 - S_k(S_{j-1}^{-1} e_1)) = e_j^* S_k^{-1}(e_1 - \underline{S}_{j-1}(S_{j-1}^{-1} e_1)) = e_j^* S_k^{-1} e_j.$$

The first equality now follows from the relation $S_k = \Gamma_k T_k \Gamma_k^{-1}$.

The matrix T_k is given by $T_k = \frac{\beta}{\alpha}(I + \frac{1}{2\beta}\Delta)$, where Δ is the k by k matrix with zeros entries everywhere except at the positions $(i-1, i)$ and $(i, i-1)$, where it has the value one and the $(2,1)$ element is 2. To obtain the estimate for $e_j^* T_k^{-1} e_j$, we express $(I + \frac{1}{2\beta}\Delta)^{-1}$ as a Neumann series and check that $e_j^*\Delta^{2i-1}e_j = 0$. With some effort it can be shown that $|e_j^*\Delta^{2i}e_j| \leq 2\frac{(2i)!}{(i!)^2}$ for all $i = 1, 2, \ldots$; see Lemma A.1 in Appendix A. Now use for $t = 1/\beta^2$ that

$$\frac{1}{\sqrt{1-t}} = \sum_{i=0}^{\infty} \frac{(2i)!}{(2^i i!)^2} t^i \quad \text{if} \quad |t| < 1.$$

This leads to the estimate in(5.5). ∎

A combination of Lemma 5.1 and (3.6) gives the following bound on the residual gap:

$$\|\mathbf{r}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_k)\|_2 \leq 2\sqrt{\mathcal{C}(\mathbf{A})}/\|\mathbf{A}\|_2 \sum_{j=0}^{k-1} \|\mathbf{f}_j\|_2 \leq 2\sqrt{\mathcal{C}(\mathbf{A})} \sum_{j=0}^{k-1} \eta_j \|\mathbf{r}_j\|_2.$$

Given the fact that we are interested in a residual precision of only $\mathcal{O}(\varepsilon)$, we propose the same relaxation strategy as for Richardson iteration in section 4, i.e., pick $\eta_j = \varepsilon/\|\mathbf{r}_j\|_2$. The gap for this strategy can then be bounded as

$$\|\mathbf{r}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_k)\|_2 \leq 2k\varepsilon\sqrt{\mathcal{C}(\mathbf{A})}. \tag{5.6}$$

The proposed relaxation strategy allows very large perturbations when the residuals are small. Nevertheless, the following theorem shows that also the initial convergence speed of the computed residuals for this strategy is close to that of the exact

method. Furthermore, the computed residuals become, in the end, sufficiently small for (5.6) to be meaningful as measure for the attainable accuracy.

THEOREM 5.2. *Let* $\bar{\mathbf{r}}_k$ *satisfy* (5.3) *with* $\eta_j = 0$, *and let* $\mathbf{r}_k$ *satisfy* (5.3) *with* $\eta_j = \varepsilon/\|\mathbf{r}_j\|_2$. *Then,*

$$\|\mathbf{r}_k - \bar{\mathbf{r}}_k\|_2 \leq \varepsilon(1 - |\gamma_k|^{-1})\mathcal{C}(\mathbf{A}).$$

Proof. If we subtract (2.3) from (3.4), then we get

$$\mathbf{A}(\mathbf{R}_k - \overline{\mathbf{R}}_k) + \mathbf{F}_k = (\mathbf{R}_{k+1} - \overline{\mathbf{R}}_{k+1})\underline{S}_k, \quad (\mathbf{R}_0 - \overline{\mathbf{R}}_0)e_1 = \mathbf{0}. \tag{5.7}$$

Let $\mathbf{v}_{\min}$ be the normalized eigenvector of $\mathbf{A}$ corresponding to $\lambda_{\min}$. We will show that $\|\bar{\mathbf{r}}_k - \mathbf{r}_k\|_2$ is maximal when for all perturbations we have $\mathbf{f}_j = \varepsilon\|\mathbf{A}\|_2\mathbf{v}_{\min}$ (or $\mathbf{F}_k = \varepsilon\|\mathbf{A}\|_2\mathbf{v}_{\min}\vec{1}^*$). Subsequently, we will solve (5.7) for these perturbations from which our claim follows.

With (2.9) we rewrite (5.7) as

$$\mathbf{A}\mathbf{D}_k + \mathbf{F}_k\Gamma_k = \mathbf{D}_{k+1}\underline{T}_k,$$

with $\mathbf{d}_j \equiv (\mathbf{r}_j - \bar{\mathbf{r}}_j)\gamma_j$. Written as a three-term recurrence this reads as

$$\mathbf{d}_j = 2\phi(\mathbf{A})\mathbf{d}_{j-1} - \mathbf{d}_{j-2} + 2\alpha\mathbf{f}_{j-1}\gamma_{j-1},$$

with $\mathbf{d}_0 = \mathbf{0}$, $\mathbf{d}_1 = \alpha\mathbf{f}_0$. This recurrence can be solved using standard techniques (e.g., [7, p. 58], [10, section 2]), which gives

$$\mathbf{d}_k = \alpha u_k(\phi(\mathbf{A}))\mathbf{f}_0\gamma_0 + \sum_{j=1}^{k-1} 2\alpha u_{k-j}(\phi(\mathbf{A}))\mathbf{f}_j\gamma_j,$$

where u_j is the so-called *Chebyshev polynomial of the second kind* (e.g., [7]), i.e., $u_{j+1}(t) = 2tu_j(t) - u_{j-1}(t)$, $u_0(t) = 0$ and $u_1(t) = 1$.

Realizing that $|u_j(t)| \leq j$ for $t \in [-1, 1]$, $u_j(-1) = (-1)^j j$ and $\operatorname{sign}(\gamma_j) = (-1)^j$ it follows that

$$\|\mathbf{d}_k\|_2 \leq \left| \varepsilon\alpha\|\mathbf{A}\|_2 \left(u_k(\phi(\lambda_{\min}))\gamma_0 + \sum_{j=1}^{k-1} 2u_{k-j}(\phi(\lambda_{\min}))\gamma_j \right) \right|.$$

This shows that the error is maximal if all perturbations are $\varepsilon\|\mathbf{A}\|_2\mathbf{v}_{\min}$.

In order to solve (5.7) with $\mathbf{F}_k = \varepsilon\|\mathbf{A}\|_2\mathbf{v}_{\min}\vec{1}^*$, we use a relation for the iterates which follows from substituting $\mathbf{R}_k = \mathbf{b}\vec{1}^* - \mathbf{A}\mathbf{X}_k$ in (2.6):

$$\mathbf{A}\mathbf{X}_k - \mathbf{b}\vec{1}^* = \mathbf{X}_{k+1}\underline{S}_k, \quad \mathbf{X}_0 e_1 = \mathbf{0}. \tag{5.8}$$

Comparing (5.8) with (5.7) shows that $\|\mathbf{r}_k - \bar{\mathbf{r}}_k\|_2$ is bounded by the norm of the $(k+1)$th approximate solution of Chebyshev iteration when the right-hand side is $\varepsilon\|\mathbf{A}\|_2\mathbf{v}_{\min}$, which is

$$\varepsilon\|\mathbf{A}\|_2\frac{1 - c_k(-1)/\gamma_k}{\lambda_{\min}}\mathbf{v}_{\min}.$$

By noting that $0 \leq c_k(-1)/\gamma_k \leq 1$ and $|c_k(-1)| = 1$ the proof can be concluded. ∎

In Figure 5.1 we give an illustration of our relaxation strategy for Chebyshev iteration similar to what we did for Richardson iteration in section 4.

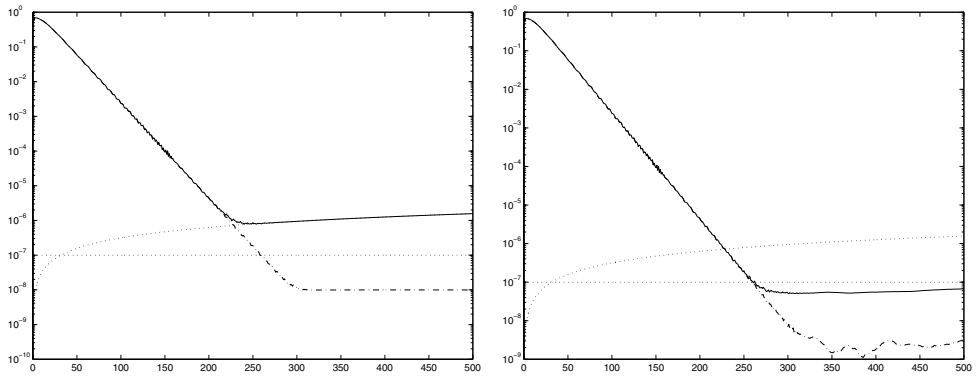


FIG. 5.1. *Chebyshev iteration with* $\eta_j = 10^{-10}/\|\mathbf{r}_j\|$, *true residuals* (—), *norm computed residual* (- ·), *and the quantities* $10^{-10}\mathcal{C}(\mathbf{A})$, $2j10^{-10}\sqrt{\mathcal{C}(\mathbf{A})}$ *(both dotted) as a function of* j. *The matrix* $\mathbf{A}$ *has dimension* 100 *and* $\mathcal{C}(\mathbf{A}) = 1000$. *Left: Errors have all components equal. Right: Random errors.*

5.1. Discussion. The effect of perturbations on the Chebyshev method has been investigated in literature. Woźniakowski analyzes in [33] the effect of finite precision arithmetic on the Chebyshev method. He describes a variant of the Chebyshev method where the residuals are computed directly and concludes that this method is forward stable. Furthermore, he points out this method is not well behaved: the residuals for this method can stagnate at a level of $\mathcal{C}(\mathbf{A})\|\mathbf{A}\|_2\,\|\mathbf{A}^{-1}\mathbf{b}\|_2$ times the machine precision. (It is interesting to note that a similar observation has been made for MINRES [28].) A method is *well behaved* if the true residuals decrease below the level of $\|\mathbf{A}\|_2\,\|\mathbf{A}^{-1}\mathbf{b}\|_2$ times the machine precision.

Gutknecht and Strakoš [20] analyze the residual gap for general Krylov subspace methods that use two three-term recurrences (one for the residuals and one for the approximate solutions). This analysis is applied in [19] in a qualitative discussion on the residual gap for the Chebyshev method. The approach from [19] differs essentially from ours in that we are using properties of the matrix $\underline{S}_k$ to bound the gap instead of a close inspection of the recursion as in [20]. The advantage is that it is easier to derive bounds in terms of global properties (as in Lemma 5.1) and our approach is not restricted to a certain type of recursion. Expressions similar to that in [20] can be obtained from (3.6) by writing out $e_j^* S_k^{-1} e_1$ using the LU-decomposition from Lemma 2.1. A difference is that, due to a different context, we do not consider perturbations on the recursion for the iterates but an analysis as in the previous sections can be easily extended to this case.

For the Chebyshev method with inexact preconditioning, called *flexible preconditioning* in this paper, convergence results have been established by Golub and Overton [10] for $\eta_j = \eta$ but where η can be modest (and much larger than ε). Moreover, under certain assumptions for the cost of the flexible preconditioner, it is shown in [9] that a fixed threshold strategy is optimal with respect to asymptotic convergence. It is not difficult to see that, if one sets the preconditioner to $\mathbf{M} = \mathbf{I}$, the residuals of this flexible process satisfy the perturbed residual relation given in (3.4). However, since the perturbation is the consequence of inexact preconditioning, instead of inexact matrix-vector products, we still have that $\mathbf{r}_j = \mathbf{b} - \mathbf{A}\mathbf{x}_j$. This shows that, although there are common elements, flexible preconditioning is different from the case of inexact matrix-vector products. Since, for the latter case, there is also an accuracy issue.

6. The inexact CG method. In this section we discuss relaxation strategies for the *CG method* [21] and some of its variants although, strictly speaking, not all variants that we discuss use gradients that are conjugate. The most popular formulation of the CG method is due to Hestenes and Stiefel [21, section 3] and consists of three coupled two-term recurrences. For $j = 1, \ldots, k$, this method, with inexact matrix-vector product, is defined by the recurrences

$$\mathbf{c} = \mathbf{A}\mathbf{p}_{j-1} + \mathbf{g}_{j-1}, \tag{6.1}$$

$$\mathbf{r}_j = \mathbf{r}_{j-1} - \alpha_{j-1}\mathbf{c}, \tag{6.2}$$

$$\mathbf{x}_j = \mathbf{x}_{j-1} + \alpha_{j-1}\mathbf{p}_{j-1}, \tag{6.3}$$

$$\mathbf{p}_j = \mathbf{r}_j + \beta_{j-1}\mathbf{p}_{j-1}, \tag{6.4}$$

with

$$\alpha_{j-1} \equiv \frac{\|\mathbf{r}_{j-1}\|_2^2}{\mathbf{p}_{j-1}^*\mathbf{c}} \quad \text{and} \quad \beta_{j-1} \equiv \frac{\|\mathbf{r}_j\|_2^2}{\|\mathbf{r}_{j-1}\|_2^2}, \tag{6.5}$$

and $\mathbf{p}_0 = \mathbf{r}_0 = \mathbf{b}$ and $\mathbf{x}_0 = \mathbf{0}$. We have added a perturbation, $\mathbf{g}_{j-1}$, to the matrix-vector product in (6.2) to obtain the inexact version with $\|\mathbf{g}_{j-1}\|_2 \leq \eta_{j-1}\|\mathbf{A}\|_2\,\|\mathbf{p}_{j-1}\|_2$.

The goal is, again, to obtain a final residual precision of about ε. Therefore, we want to investigate the influence of the η_j on the residual gap and we make the assumption that the computed residuals become sufficiently small in the end as for Chebyshev iteration in the previous section.

We define

$$\widetilde{U}_k \equiv \begin{bmatrix} 1 & -\beta_0 & & & \\ & 1 & -\beta_1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & -\beta_{k-2} \\ & & & & 1 \end{bmatrix}, \Delta_k \equiv \begin{bmatrix} \alpha_0 & & & & \\ & \alpha_1 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \alpha_{k-1} \end{bmatrix}.$$

This gives us the following equivalent matrix formulations of the recurrences of the inexact CG method:

$$\mathbf{A}\mathbf{P}_k + \mathbf{G}_k = \mathbf{R}_{k+1}\underline{J}_k\Delta_k^{-1}, \quad \mathbf{X}_{k+1}\underline{J}_k = -\mathbf{P}_k\Delta_k, \quad \mathbf{R}_k = \mathbf{P}_k\widetilde{U}_k.$$

Combining these relations shows that

$$\mathbf{A}\mathbf{R}_k + (\mathbf{G}_k\widetilde{U}_k) = \mathbf{R}_{k+1}(\underline{J}_k\Delta_k^{-1}\widetilde{U}_k) \text{ and } -\mathbf{R}_k = \mathbf{X}_{k+1}(\underline{J}_k\Delta_k^{-1}\widetilde{U}_k). \tag{6.6}$$

We see that (3.4) and (2.6) are satisfied for this method with $\underline{S}_k \equiv \underline{J}_k\Delta_k^{-1}\widetilde{U}_k$ and $\mathbf{F}_k \equiv \mathbf{G}_k\widetilde{U}_k$. Therefore, we can use our familiar formula (3.6) to get an expression for the residual gap:

$$\mathbf{r}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_k) = -\mathbf{F}_k S_k^{-1} e_1 = -\mathbf{G}_k\widetilde{U}_k S_k^{-1} e_1 = -\mathbf{G}_k\Delta_k J_k^{-1} e_1 = -\sum_{j=0}^{k-1} \alpha_j \mathbf{g}_j.$$

This expression can also be obtained by an inductive combination of (6.2) and (6.3). This simpler argument, that avoids the matrix formulation, was used in [27, 15].

However, the present argument explains how CG fits in the general framework of this paper. Moreover, for the conclusions below we need the matrix formulation anyway.

From $\|\mathbf{g}_j\|_2 \leq \eta_j \|\mathbf{A}\|_2 \|\mathbf{p}_j\|_2$, we get the following bound on the norm of the residual gap:

$$\|\mathbf{r}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_k)\|_2 \leq \sum_{j=0}^{k-1} \eta_j |\alpha_j| \|\mathbf{A}\|_2 \|\mathbf{p}_j\|_2. \tag{6.7}$$

Thus, the problem of deriving relaxation strategies for the CG method amounts to bounding $|\alpha_j| \|\mathbf{p}_j\|_2$. We do this in the remainder of this section.

The CG method is intimately connected with the Lanczos method, e.g., [11, Chapter 9]. In order to continue we introduce for theoretical purposes the following *inexact* Lanczos process:

$$\mathbf{A}\mathbf{V}_k + \widetilde{\mathbf{F}}_k = \mathbf{V}_{k+1} \underline{T}_k, \tag{6.8}$$

where $\underline{T}_k \equiv \Gamma_{k+1}^{-1} \underline{S}_k \Gamma_k$, $\Gamma_k \equiv \operatorname{diag}(\vec{\gamma}_{k-1})$, $\gamma_j \equiv (-1)^j \|\mathbf{r}_j\|_2^{-1}$, $\mathbf{V}_k \equiv \mathbf{R}_k \Gamma_k$, and $\widetilde{\mathbf{F}}_k \equiv \mathbf{F}_k \Gamma_k$. From (6.6) and Section 2 it follows that $\mathbf{x}_j = \mathbf{R}_j S_j^{-1} e_1 = \mathbf{V}_j T_j^{-1} e_1$ and combining this with (6.3) shows that

$$\alpha_j \mathbf{p}_j = \mathbf{V}_k (T_{j+1}^{-1} e_1 - T_j^{-1} e_1). \tag{6.9}$$

We will use this relation to bound $|\alpha_j| \|\mathbf{p}_j\|_2$.

6.1. The case of T_k positive definite. First we assume that T_k is positive definite. In the previous section we reduced the problem of bounding the gap to bounding $|\alpha_j| \|\mathbf{p}_j\|_2$. We will do this using (6.9) and the following result.

LEMMA 6.1. *Let $j < k$. Then,*

$$T_{j+1}^{-1} e_1 - T_j^{-1} e_1 = T_{j+1}^{-1} \frac{e_{j+1}}{\gamma_j} = \frac{\vec{\gamma}_j}{\vec{\gamma}_j^* T_{j+1} \vec{\gamma}_j}. \tag{6.10}$$

Proof. First observe that

$$T_{j+1}^{-1} e_1 - T_j^{-1} e_1 = T_{j+1}^{-1}(e_1 - T_{j+1} T_j^{-1} e_1) = T_{j+1}^{-1}(e_1 - \underline{T}_j T_j^{-1} e_1).$$

Now, the first identity in (6.10) follows from Lemma 2.2.

Since $\vec{\gamma}_{j+1}^* \underline{T}_{j+1} = \vec{0}^*$, we see that $\vec{\gamma}_j^* T_{j+1} = \delta e_{j+1}^*$ for some scalar δ. Multiplication from the right with $\vec{\gamma}_j$ shows that $\delta = \vec{\gamma}_j^* T_{j+1} \vec{\gamma}_j / \gamma_j$. Since T_{j+1} is symmetric, we find $\vec{\gamma}_j = \delta T_{j+1}^{-1} e_{j+1}$, which leads to the second identity. ∎

We combine this lemma with (6.9) and arrive at the estimate

$$|\alpha_j| \|\mathbf{p}_j\|_2 \leq \|\mathbf{V}_k\|_2 \|T_k^{-1}\|_2 \rho_j, \text{ with } \rho_j \equiv \frac{1}{\|\vec{\gamma}_j\|_2} = \left(\sum_{i=0}^{j} \|\mathbf{r}_i\|_2^{-2} \right)^{-1/2}. \tag{6.11}$$

Inserting this estimate in (6.7), we find the following bound on the norm of the residual gap:

$$\|\mathbf{r}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_k)\|_2 \leq \|\mathbf{V}_k\|_2 \|\mathbf{A}\|_2 \|T_k^{-1}\|_2 \sum_{j=0}^{k-1} \eta_j \rho_j. \tag{6.12}$$

This estimate can be further bounded using that $\|\mathbf{V}_k\|_2 \leq \|\mathbf{V}_k\|_F \leq \sqrt{k}$. In practice, this turns out to be crude since $\|\mathbf{V}_k\|_2$ is close to one or only a modest multiple of one. If $\mathbf{A}$ is symmetric positive definite, then, in the exact case, $\|T_k^{-1}\|_2 \leq \|\mathbf{A}^{-1}\|_2$. In the inexact case, $\|\mathbf{A}\|_2\|T_k^{-1}\|_2$ can be viewed as an approximation to $\mathcal{C}(\mathbf{A})$. It is tempting to refer to the results of Paige [23] for perturbed Lanczos processes to bound this quantity. However, the perturbations in our context are not assumed to be uniformly bounded. In fact, they are allowed to grow during the process. Therefore, we cannot make use of his results. Of course, we can monitor this quantity during the inexact process and, possibly, incorporate this estimate into our tolerance η_j.

Bouras, Frayssé, and Giraud proposed in [4], following their work for inexact GMRES and (3.3), a relaxation strategy for the CG method where they take

$$\eta_j = \max\left\{\frac{\varepsilon}{\|\mathbf{r}_j\|_2}, \varepsilon\right\}. \tag{6.13}$$

If we take the larger tolerance $\eta_j = \varepsilon/\rho_j$ (since $\rho_j \leq \|\mathbf{r}_j\|_2$), then we have from (6.12) that

$$\|\mathbf{r}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_k)\|_2 \leq \varepsilon k \|\mathbf{V}_k\|_2 \|\mathbf{A}\|_2 \|T_k^{-1}\|_2. \tag{6.14}$$

We saw that our analysis of the residual gap helps to provide insight into the practical success of the Bouras–Frayssé–Giraud condition (6.13) and even suggests that we can relax stronger than previously proposed. Indeed, numerical experiments with symmetric positive definite matrices $\mathbf{A}$ confirm this.

An alternative for bounding $|\alpha_j|\|\mathbf{p}_j\|_2$ follows from noticing that in (6.10), for a fixed value of i, the quantities $e_i^*(T_j^{-1}e_1 - T_{j-1}^{-1}e_1)$ have a constant sign for all j (or are zero). Therefore, we have that

$$\|T_j^{-1}e_1 - T_{j-1}^{-1}e_1\|_2 \leq \|T_i^{-1}e_1\|_2 \quad \text{for} \quad i \geq j.$$

This provides a similar bound on $|\alpha_j|\|\mathbf{p}_j\|_2$ as derived by Greenbaum in [15] for the residual gap of CG in order to study the attainable accuracy of the CG method in finite precision computations. She uses that the errors of the CG method are monotonically decreasing in 2-norm in order to bound $\|\alpha_j\mathbf{p}_j\|_2$. In our context this approach is too crude since it does not lead to a relaxation strategy.

6.2. The case of T_k indefinite. The CG method is still used in practice for solving Hermitian indefinite systems, despite its lack of robustness. One reason is that, although the tridiagonal matrix can be ill conditioned in one iteration, this can never happen for two consecutive iterations, e.g., [1, 17]. If $\mathbf{A}$ is symmetric indefinite but nonsingular, then, even in the exact case, T_k will not be definite and we cannot uniformly bound $\vec{\gamma}_j^* T_k \vec{\gamma}_j$ away from zero. We may not expect that Lemma 6.1 leads to useful results for bounding $|\alpha_j|\|\mathbf{p}_j\|_2$ using (6.9). As an alternative, we use the following lemma.

LEMMA 6.2. *Let $j < k$. Then,*

$$T_{j+1}^{-1}e_1 - T_j^{-1}e_1 = \underline{T}_{j+1}^{\dagger}\left(\frac{e_{j+1}}{\gamma_j} - \frac{e_{j+2}}{\gamma_{j+1}}\right). \tag{6.15}$$

Proof. We observe that $\underline{T}_{j+1}^{\dagger}\underline{T}_{j+1}$ is the identity on $j+1$-vectors and conclude that

$$T_{j+1}^{-1}e_1 - T_j^{-1}e_1 = \underline{T}_{j+1}^{\dagger}\left((e_1 - \underline{T}_{j+1}T_j^{-1}e_1) - (e_1 - \underline{T}_{j+1}T_{j+1}^{-1}e_1)\right).$$

The proof can be concluded by rewriting the expressions on the right with the help of Lemma 2.2. ∎

If we use that $\|\underline{T}_{j+1}^{\dagger}\|_2 \leq \|\underline{T}_k^{\dagger}\|_2$ for $k > j$ and, from (6.5), that $\beta_j = \gamma_j^2/\gamma_{j+1}^2$, then we can bound the norm of the residual gap as

$$\|\mathbf{r}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_k)\|_2 \leq \|\mathbf{V}_k\|_2 \|\mathbf{A}\|_2 \|\underline{T}_k^{\dagger}\|_2 \sum_{j=0}^{k-1} \eta_j \|\mathbf{r}_j\|_2 \sqrt{1+\beta_j}. \tag{6.16}$$

A similar expression can be found in [27, 15], where the perturbations are assumed to be small and second order terms have been neglected (then it can be proven that $\|\mathbf{A}\|_2 \|\underline{T}_k^{\dagger}\|_2 \lesssim \mathcal{C}(\mathbf{A})$). For the choice $\eta_j = \varepsilon/\|\mathbf{r}_j\|_2$, we get, using (6.16),

$$\|\mathbf{r}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_k)\|_2 \leq \varepsilon k \|\mathbf{V}_k\|_2 \|\mathbf{A}\|_2 \|\underline{T}_k^{\dagger}\|_2 \max_{0 \leq j < k} \sqrt{1+\beta_j}. \tag{6.17}$$

We see that, as long as the β_j are bounded, this strategy can work very well. However, practical problems often lead to a matrix $\mathbf{A}$ that is indefinite, for instance in the QCD example discussed in section 3. In this case there can be very large intermediate residuals caused by an eigenvalue of T_k being "accidentally" close to zero. The situation of an eigenvalue of T_k close to zero is in literature often referred to as a *near breakdown*. It results in a value of β_j that is very large, and it follows from (6.17) that the proposed strategy in (6.13) may fail in achieving the required residual precision.

From (6.16) it follows that picking $\eta_j = \varepsilon/(\|\mathbf{r}_{j+1}\|_2 + \|\mathbf{r}_j\|_2)$ is a better strategy in this case. However, this is not practical since the size of $\mathbf{r}_{j+1}$ is not known yet. An alternative is to consider the first bound in (6.7) and pick

$$\eta_j = \frac{\varepsilon}{|\alpha_j| \|\mathbf{p}_j\|_2}.$$

If the approximation of the matrix-vector product is computed with an iterative method, then the inner product of $\mathbf{p}_j$ with the "current" approximation to the matrix-vector product can be monitored (at the cost of an additional inner product), and from this α_j can be estimated. Nevertheless, in case of a near breakdown a very accurate matrix-vector product is still necessary. We will therefore consider variants of the CG method in Section 6.4.

6.3. The behavior of the computed residuals. Studying the convergence and stagnation level of the computed residuals is a much more difficult topic. Greenbaum [14] showed that the convergence of a slightly perturbed CG process is equal to that of the exact method applied to a matrix with eigenvalues in small clusters around the eigenvalues of the original matrix. The width of these clusters is determined by the size of the perturbation of the Lanczos process. Unfortunately, this analysis does not apply in our situation since it does not explain why the accuracy of the matrix-vector product can be relaxed when the CG method converges as was the case for Richardson iteration and Chebyshev iteration in the previous sections. Numerical experiments indeed suggest that a relaxation strategy for the accuracy of the matrix-vector products does not spoil the convergence of the computed residuals and they seem to stagnate at a level in the order of ε.

However, the convergence speed can be very different from that of the exact CG method. It is important to mention that in numerical experiments we observe that a near breakdown of the method can severely alter the behavior of the computed

residuals. In this case, $\widetilde{\mathbf{F}}_k$ in (6.8) has some relatively very large columns. To see this we mention that for the jth column of $\widetilde{\mathbf{F}}_k$ we have that $\|\widetilde{\mathbf{f}}_{j-1}\| = \|\mathbf{g}_{j-1} - \beta_{j-2}\,\mathbf{g}_{j-2}\|_2/\|\mathbf{r}_{j-1}\|_2$. A simple analysis shows that

$$\|\mathbf{p}_{j-1}\|_2 = \|\mathbf{R}_k\widetilde{U}_k^{-1}e_j\|_2 \le \|\mathbf{R}_k\Gamma_k\|_2\,\|\Gamma_k^{-1}\widetilde{U}_k^{-1}e_j\|_2 = \|\mathbf{V}_k\|_2\frac{\|\mathbf{r}_{j-1}\|_2^2}{\rho_{j-1}},$$

where ρ_j is as defined in (6.11). Notice that ρ_j can be viewed as the norm of a smoothed residual, e.g., [21, Section 7]. We have the following upper bound for the norm of the jth column of $\widetilde{\mathbf{F}}_k$:

$$\|\widetilde{\mathbf{f}}_{j-1}\| = \|\mathbf{g}_{j-1} - \beta_{j-2}\mathbf{g}_{j-2}\|_2/\|\mathbf{r}_{j-1}\|_2 \le \|\mathbf{A}\|_2\,\|\mathbf{V}_k\|_2\,\|\mathbf{r}_{j-1}\|_2\left(\frac{\eta_{j-1}}{\rho_{j-1}} + \frac{\eta_{j-2}}{\rho_{j-2}}\right).$$

The ratio $\|\mathbf{r}_{j-1}\|_2/\rho_{j-1}$ is large in case of a near breakdown since then we have that $\rho_{j-1} \ll \|\mathbf{r}_{j-1}\|_2$. This shows that when there is a near breakdown, there can be a relatively very large perturbation of the Lanczos relation. One consequence is a large residual gap (as discussed). Another effect is a potential delay in the convergence (or even worse). A simple numerical example is given in the next section.

6.4. Variants of the CG method. Mathematically equivalent variants of the CG method can be derived from the Lanczos method. In this section we will consider two such alternatives. These methods are based on a three-term recurrence for the residuals instead of the coupled two-term recurrences of the Hestenes and Stiefel implementation discussed in the previous sections. We start with a short derivation of these alternatives.

Since the CG residuals are multiples of the Lanczos vectors, we can derive the coefficients for the recurrence (2.3) from the Lanczos relation by virtue of (2.9). To see this, we write

$$\underline{T}_k \equiv \begin{bmatrix} \alpha_0 & \beta_0 & & & \\ \beta_0 & \alpha_1 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{k-2} \\ & & & \beta_{k-2} & \alpha_{k-1} \\ & & & & \beta_{k-1} \end{bmatrix}, \quad \underline{S}_k \equiv \begin{bmatrix} \mu_0 & \delta_0 & & & \\ \tau_0 & \mu_1 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \delta_{k-2} \\ & & & \tau_{k-2} & \mu_{k-1} \\ & & & & \tau_{k-1} \end{bmatrix}.$$

The matrix $\underline{T}_k$ is computed using the Lanczos method and we want expressions for the elements of the matrix $\underline{S}_k$. We can do this similar to our derivation of Chebyshev iteration in Section 5. From the necessary property that $\vec{1}^*\underline{S}_k = \vec{0}$, it immediately follows that $\tau_j = -(\mu_j + \delta_{j-1})$ (with $\delta_{-1} = 0$). Using (2.9) we see that $\mu_j = \alpha_j$, $\delta_j = \beta_j(\gamma_j/\gamma_{j+1})$ and $\tau_j = \beta_j(\gamma_{j+1}/\gamma_j)$. Eliminating β_j gives that $\delta_j = \tau_j(\gamma_j/\gamma_{j+1})^2$. With $\delta_{-1} = 0$ we get, using Lemma 2.2,

$$\delta_j = \tau_j\frac{\|\mathbf{r}_{j+1}\|_2^2}{\|\mathbf{r}_j\|_2^2}, \quad \mu_j = \frac{\mathbf{r}_j^*\mathbf{A}\mathbf{r}_j}{\|\mathbf{r}_j\|_2^2}, \quad \tau_j = -(\mu_j + \delta_{j-1}).$$

Computing the residuals and iterates with these coefficients and the recurrences given in (2.3) and (2.6) gives a variant of CG known as *Orthores* (where we use the nomenclature from [20]).

Rutishauser's variant of this method is obtained by introducing auxiliary variables $\Delta\mathbf{x}_j$ and $\Delta\mathbf{r}_j$ using the LU-decomposition, $\underline{S}_k = \underline{J}_k U_k$, from Lemma 2.1 where $(U_k)_{j,j} = -\tau_{j-1}$ and $(U_k)_{j+1,j} = \delta_{j-1}$. This gives

$$\begin{aligned} \mathbf{R}_{k+1}\underline{J}_k &= \Delta\mathbf{R}_k, \quad \Delta\mathbf{R}_k\, U_k = \mathbf{A}\mathbf{R}_k \quad \text{and} \\ \mathbf{X}_{k+1}\underline{J}_k &= \Delta\mathbf{X}_k, \quad \Delta\mathbf{X}_k\, U_k = -\mathbf{R}_k. \end{aligned} \tag{6.18}$$

Now that we have defined the two methods, we shift our attention to the inexact case. In *inexact* Orthores the matrix-vector product is perturbed in step j with a term $\mathbf{g}_{j-1}$. This leads to the (familiar) perturbed residual relation

$$A\mathbf{R}_k + \mathbf{F}_k = \mathbf{R}_{k+1}\underline{S}_k, \quad \text{with} \quad \mathbf{R}_k e_1 = \mathbf{b}, \quad \vec{1}^*\underline{S}_k = \vec{0}^*,$$

where $\mathbf{F}_k = \mathbf{G}_k$ and, therefore, $\|\mathbf{f}_j\|_2 \leq \eta_j \|\mathbf{A}\|_2 \|\mathbf{r}_j\|_2$. For the inexact version of Rutishauser's method we have $\Delta\mathbf{R}_k\, U_k = \mathbf{A}\mathbf{R}_k + \mathbf{G}_k$, and it follows that, for the same perturbations, the inexact version of Orthores and Rutishauser's variant are equivalent under the assumption of exact arithmetic and, hence, the same upper bounds apply.

We want to bound the gap for the discussed methods and derive a suitable relaxation strategy. Therefore, we notice that the residuals of inexact Orthores are now multiples (γ_j^{-1}) of the Lanczos vectors of an inexact Lanczos process given by (6.8) with $\underline{T}_k \equiv \Gamma_{k+1}^{-1}\underline{S}_k\Gamma_k$, $\Gamma_k \equiv \operatorname{diag}(\vec{\gamma}_{k-1})$ and $\gamma_j \equiv (-1)^j \|\mathbf{r}_j\|_2^{-1}$. Combining this with Lemma 3.1 shows that

$$|e_j^* S_k^{-1} e_1| \leq \|\underline{T}_k^\dagger\|_2 \frac{1}{\|\mathbf{r}_{j-1}\|_2} \left(\rho_{j-1} + \|\mathbf{r}_k\|_2\right), \tag{6.19}$$

where ρ_{j-1} is as defined in (6.11). The general expression for the residual gap (3.6), now leads to the following bound:

$$\begin{aligned} \|\mathbf{r}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_k)\|_2 &\leq \|\underline{T}_k^\dagger\|_2 \sum_{j=0}^{k-1} \|\mathbf{r}_j\|_2^{-1} \left(\rho_j + \|\mathbf{r}_k\|_2\right) \|\mathbf{f}_j\|_2 \\ &\leq \|\mathbf{A}\|_2 \|\underline{T}_k^\dagger\|_2 \sum_{j=0}^{k-1} \eta_j \left(\rho_j + \|\mathbf{r}_k\|_2\right). \end{aligned}$$

Recall that we assume that the computed residuals ultimately become small enough. Now, assume that we terminate the iterative process for $\|\mathbf{r}_k\|_2 \leq \varepsilon$. In this case we see that the size of the gap is essentially determined by the values of the ρ_j, the η_j, and $\|\underline{T}_k^\dagger\|_2$. Unfortunately, we have no a priori knowledge about the size of $\|\underline{T}_k^\dagger\|_2$. We hope that this quantity is in the order of $\|\mathbf{A}^{-1}\|_2$. For inexact Orthores (and Rutishauser's variant) we propose the following relaxation strategy:

$$\eta_j = \frac{\varepsilon}{\rho_j}, \tag{6.20}$$

where ρ_j is given in (6.19) and can be computed at little additional cost. For the proposed relaxation strategy in (6.20), we have for the residual gap

$$\|\mathbf{r}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_k)\|_2 \leq \varepsilon k \|\mathbf{A}\|_2 \|\underline{T}_k^\dagger\|_2 \left(1 + \frac{\|\mathbf{r}_k\|_2}{\rho_k}\right).$$

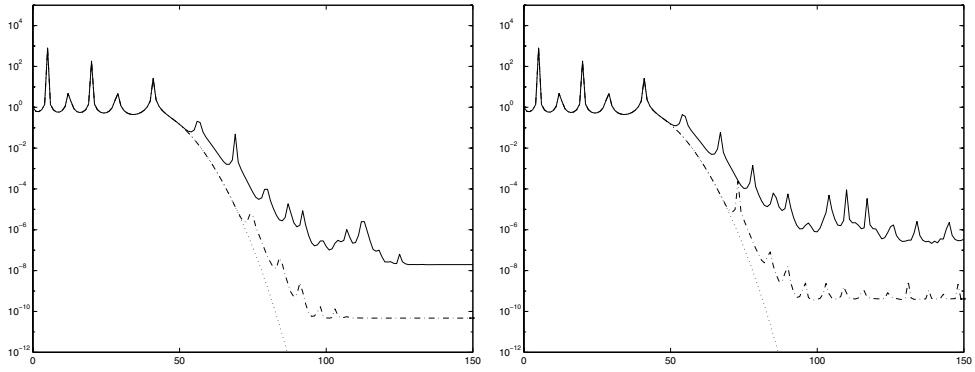


FIG. 6.1. *True residuals exact FOM (dotted), CG (–), Orthores (- ·), Rutishauser's variant (dots) as a function of j. In both pictures $\varepsilon = 10^{-10}$. Left: $\eta_j = \varepsilon$. Right: $\eta_j = \varepsilon/\rho_j$.*

This shows that the distance between the computed and true residual can be large when there is a near breakdown but when the process is terminated, if $\|\mathbf{r}_k\|_2 \leq \varepsilon$, the gap is hopefully $\mathcal{O}(\varepsilon)$. An alternative is to pick $\eta_j = \varepsilon/(\varepsilon + \rho_j)$ which somewhat simplifies the resulting expression that bounds the gap.

Let us summarize our findings. If we consider the upper bounds on the residual gap, we see that for the two discussed variants based on a three-term recurrence there is no need in computing the matrix-vector product more accurately in case of a near breakdown in contrast to the standard coupled two-term based recurrence implementation of CG. As seen, we can exploit this in our relaxation strategy. For indefinite matrices $\mathbf{A}$, where the convergence behavior of the residuals is highly irregular, the alternative CG methods and relaxation strategy in this section can offer advantages over CG and the relaxation strategy by Bouras, Frayssé, and Giraud in (6.13). Furthermore, for the three-term recurrences, a near breakdown does not lead to a large perturbation of the (implicit) Lanczos relation. Hence, we expect the effect of loss of convergence speed caused by near breakdowns less dramatic than for CG.

In Figure 6.1 we give a simple illustration. The right-hand side has all components equal and the matrix is $\mathbf{A} = \mathrm{diag}(1:100) - 5.2025\,\mathbf{I}$. The shift causes a large intermediate residual in the fifth step. The figure illustrates that Orthores and Rutishauser's variant perform equal and better than the CG method with respect to accuracy and convergence speed. Here, we prefer to use the three-term recurrence variants over the coupled two-term recurrences.

6.5. Discussion. For positive definite systems, the standard CG method seems appropriate in the inexact setting. The observations in the previous section show that (in the inexact setting) the use of a three-term recurrence for solving Hermitian indefinite systems can offer advantages over the standard CG implementation, especially in situations where the matrix $\mathbf{A}$ is not too ill-conditioned and convergence is irregular. Numerical experiments are given in section 8.

Numerical experiments (not reported here) suggest that this is not necessarily the case when floating point errors are the only source of errors. For example, near-breakdowns also influence the attainable precision of Rutishauser's variant of the CG method, just as for standard CG. Orthores, on the other hand, seems not sensitive to peaks but appears to be [20], like Chebyshev iteration and MINRES, not well behaved (cf. section 5.1). Our analysis can be extended for making a rounding error analysis

of several variants of the CG method for indefinite systems. This can help identify the different design choices in the construction of a CG method that influence the accuracy.

Studying the behavior of the computed residuals is a much more difficult subject. In general we observe in numerical experiments that the computed residuals become small enough for the residual gap to be a meaningful indicator for the attainable residual precision. It is also often observed that the initial convergence speed is comparable to the convergence speed of the exact method. Nevertheless, in a few cases, small perturbations of the matrix-vector product can delay convergence for the CG method and its variants. This also is the case for inexact GMRES that we discuss in the next section and we refer to this section for a numerical example and further discussion.

As a final remark we notice that we could have proposed inexact MINRES as the alternative for indefinite systems. We have not done this here for two reasons. A simple analysis of inexact MINRES shows that essentially the same bound applies as for inexact Orthores, and therefore the same relaxation strategy is appropriate. Second, we want to illustrate that the underlying mechanism for constructing the Krylov subspace is important and *not* the chosen optimality properties of the residuals. This is also illustrated in the next section in our discussion about inexact FOM and GMRES.

7. Inexact FOM and GMRES. The Lanczos method is a starting point for the derivation of a large class of iterative methods for Hermitian matrices $\mathbf{A}$. For non-Hermitian systems, the *Arnoldi* method (see, for instance, [11, section 9.4]) can be used for constructing an orthonormal basis $\mathbf{v}_0, \ldots, \mathbf{v}_k$ for $\mathcal{K}_{k+1}$ and can therefore serve as a starting point. The Arnoldi method can be summarized by the following relation:

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\underline{T}_k, \quad \mathbf{V}_k e_1 = \mathbf{b}, \tag{7.1}$$

where $\underline{T}_k$ is $k+1$ by k upper Hessenberg and $\mathbf{V}_k$ is n by k and orthogonal. Recall that $\mathbf{b}$ is assumed to have unit length.

If in step j of the Arnoldi method the matrix-vector product is computed approximately, i.e., a perturbation $\mathbf{g}_{j-1}$ is added to the matrix-vector product $\mathbf{A}\mathbf{v}_{j-1}$, then we obtain an *inexact* Arnoldi method. This latter method satisfies the following perturbed Arnoldi relation:

$$\mathbf{A}\mathbf{V}_k + \widetilde{\mathbf{F}}_k = \mathbf{V}_{k+1}\underline{T}_k, \quad \mathbf{V}_k e_1 = \mathbf{b}, \tag{7.2}$$

where $\widetilde{\mathbf{F}}_k = \mathbf{G}_k$ and, therefore, $\|\widetilde{\mathbf{f}}_j\| \leq \eta_j \|\mathbf{A}\|_2 \, \|\mathbf{v}_j\|_2 = \eta_j \|\mathbf{A}\|_2$. An interesting observation is that $\mathbf{V}_k$ is still an orthogonal matrix, but now the columns span the Krylov subspace $\mathcal{K}_k(\widehat{\mathbf{A}}_k, \mathbf{b})$ with $\widehat{\mathbf{A}}_k \equiv \mathbf{A} + \widetilde{\mathbf{F}}_k \mathbf{V}_k^*$. We will assume in this section that T_j is invertible and $\underline{T}_j$ has full rank for $j \leq k$.

The *inexact* FOM and *inexact* GMRES method [3] use the Arnoldi relation explicitly and construct their iterates as

$$y_j^{\mathrm{F}} = T_j^{-1} e_1, \quad \mathbf{x}_j^{\mathrm{F}} = \mathbf{V}_j y_j^{\mathrm{F}} \quad \text{and} \quad y_j^{\mathrm{G}} = \underline{T}_j^{\dagger} e_1, \quad \mathbf{x}_j^{\mathrm{G}} = \mathbf{V}_j y_j^{\mathrm{G}}.$$

The corresponding computed residuals are given by

$$\mathbf{r}_j^{\mathrm{F}} = \mathbf{V}_{j+1}(I - \underline{T}_j T_j^{-1}) e_1 \quad \text{and} \quad \mathbf{r}_j^{\mathrm{G}} = \mathbf{V}_{j+1}(I - \underline{T}_j \underline{T}_j^{\dagger}) e_1.$$

These expressions are a special case of (2.10) and (2.12) and, therefore, we get from Lemma 2.2 that $\mathbf{r}_j^{\mathrm{F}} = \mathbf{v}_j/\gamma_j$ and $\mathbf{r}_j^{\mathrm{G}} = \|\vec{\gamma}_j\|_2^{-2}\mathbf{V}_j\vec{\gamma}_j$, where $\vec{\gamma}_k$ is as defined in Section 2, i.e., $\gamma_k^*\underline{T}_k = \vec{0}^*$ and $\vec{\gamma}_k^* e_1 = 1$. This gives the following relation between the norms of the computed residuals of inexact FOM and inexact GMRES:

$$\rho_j \equiv \|\mathbf{r}_j^{\mathrm{G}}\|_2 = \left(\sum_{i=0}^{j} \|\mathbf{r}_i^{\mathrm{F}}\|^{-2}\right)^{-1/2}. \tag{7.3}$$

The same result is well known for exact FOM and exact GMRES from the work of Brown [5].

Notice that an alternative expression for the residuals is given by $\mathbf{r}_j^{\mathrm{F}} = \mathbf{b} - \widehat{\mathbf{A}}_j\mathbf{x}_j^{\mathrm{F}}$ and similarly for inexact GMRES. Hence, inexact FOM/GMRES is equivalent to exact (or ideal) FOM/GMRES applied to the linear system $\widehat{\mathbf{A}}_n\mathbf{x} = \mathbf{b}$. Therefore, these methods, after at most n steps, terminate with $\mathbf{x}_n^{\mathrm{F}} = \mathbf{x}_n^{\mathrm{G}} = (\mathbf{A} + \widetilde{\mathbf{F}}_n\mathbf{V}_n^*)^{-1}\mathbf{b}$ and in the inexact GMRES method, the computed residuals are monotonically decreasing. In the remainder of this section, we will drop the superscripts F or G in expressions that are valid for both methods.

In order to bound the residual gap in step k, we use an expression for the gap that is equivalent to (3.6) but is expressed in terms of the matrix $\widetilde{\mathbf{F}}_k$ (this simplifies the analysis in this section somewhat). We have

$$\mathbf{r}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_k) = \mathbf{r}_k - (\mathbf{b} - (\widehat{\mathbf{A}}_k - \widetilde{\mathbf{F}}_k\mathbf{V}_k^*)\mathbf{x}_k) = -\widetilde{\mathbf{F}}_k y_k. \tag{7.4}$$

Hence,

$$\|\mathbf{r}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_k)\|_2 = \|\widetilde{\mathbf{F}}_k y_k\|_2 \le \|\mathbf{A}\|_2 \sum_{j=0}^{k-1} \eta_j |e_{j+1}^* y_k|. \tag{7.5}$$

Since the iterates of inexact FOM and GMRES ultimately will approach the same vector $\widehat{\mathbf{A}}_n^{-1}\mathbf{b}$, and thus $y_k^{\mathrm{F}} \approx y_k^{\mathrm{G}}$, it is evident from (7.4) that an appropriate relaxation strategy for inexact GMRES is also suitable for inexact FOM, and vice versa. This will be confirmed by the analysis below.

If we plug (3.7) into (7.5), then we get the following bound for the residual gap of inexact FOM,

$$\|\mathbf{r}_k^{\mathrm{F}} - (\mathbf{b} - \mathbf{A}\mathbf{x}_k^{\mathrm{F}})\|_2 \le \|\mathbf{A}\|_2 \, \|\underline{T}_k^\dagger\|_2 \sum_{j=0}^{k-1} \eta_j (\|\mathbf{r}_j^{\mathrm{G}}\|_2 + \|\mathbf{r}_k^{\mathrm{F}}\|_2), \tag{7.6}$$

and for inexact GMRES we get

$$\|\mathbf{r}_k^{\mathrm{G}} - (\mathbf{b} - \mathbf{A}\mathbf{x}_k^{\mathrm{G}})\|_2 \le \|\mathbf{A}\|_2 \, \|\underline{T}_k^\dagger\|_2 \sum_{j=0}^{k-1} \eta_j \|\mathbf{r}_j^{\mathrm{G}}\|_2. \tag{7.7}$$

We follow the same approach as for Orthores in section 6.4 and assume that we terminate the inexact FOM/GMRES method in step k when $\|\mathbf{r}_k\|_2 \le \varepsilon$, where ε is again in the order of the required residual precision. We see that in step k the residual gap is essentially determined by the tolerances η_j, the $\|\mathbf{r}_j^{\mathrm{G}}\|_2$ (or ρ_j), and the smallest singular value of $\underline{T}_k$. Again, the size of the smallest singular value of the

Hessenberg matrix is difficult to estimate a priori (we can, however, monitor it during the iterations and incorporate this quantity in our choice for η). We, again, see that relaxation is possible with $\eta_j = \varepsilon/\rho_j$. This results for inexact FOM in the bound

$$\|\mathbf{r}_k^{\mathrm{F}} - (\mathbf{b} - \mathbf{A}\mathbf{x}_k^{\mathrm{F}})\|_2 \leq \varepsilon k \|\mathbf{A}\|_2 \|\underline{T}_k^{\dagger}\|_2 \left(1 + \frac{\|\mathbf{r}_k^{\mathrm{F}}\|_2}{\rho_k}\right), \tag{7.8}$$

and for inexact GMRES we get

$$\|\mathbf{r}_k^{\mathrm{G}} - (\mathbf{b} - \mathbf{A}\mathbf{x}_k^{\mathrm{G}})\|_2 \leq \varepsilon k \|\mathbf{A}\|_2 \|\underline{T}_k^{\dagger}\|_2. \tag{7.9}$$

We see that the relaxation strategy derived from the bounds on the residual gap confirms the empirical choice of Bouras and Frayssé in (3.3) for GMRES and can explain the success of this approach. See also the numerical experiments in [3]. Furthermore, we note that the expression for the residual gap of the inexact FOM method and inexact Orthores from the previous section coincide which can be explained by the fact that, for both methods, the matrix-vector products in the exact counterparts are applied to an orthogonal basis. Of course, the behavior of the computed residuals and the values of $\|\underline{T}_k^{\dagger}\|_2$ differ.

7.1. The behavior of the computed residuals. For inexact GMRES we know that the size of the computed residuals monotonically decrease and $\mathbf{r}_n = \mathbf{0}$. Therefore the gap provides, in the end, useful information about the attainable accuracy. However, this does not say anything about the speed of convergence of the perturbed process. The many numerical experiments in [3] suggest that the convergence of the inexact method with the proposed relaxation strategy is comparable to the convergence speed of the exact method. It is, however, very difficult to give a rigorous analysis of this observation. In some cases it can be proven that convergence of the relaxed process is approximately as fast as for the unperturbed process (similar to what we have seen for Chebyshev iteration). This is, for example, the case for inexact processes where the perturbation is of the special form

$$\widetilde{\mathbf{F}}_k = \mathbf{V}_{k+1}\underline{E}_k, \tag{7.10}$$

with $\underline{E}_k$ some upper Hessenberg matrix. In this case we have

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\underline{T}_k - \widetilde{\mathbf{F}}_k = \mathbf{V}_{k+1}\overline{\underline{T}}_k, \quad \text{with} \quad \overline{\underline{T}}_k \equiv \underline{T}_k - \underline{E}_k.$$

This shows that only the Hessenberg matrix $\underline{T}_k$ differs from the Hessenberg matrix of the unperturbed process $\overline{\underline{T}}_k$ and the perturbation does not change the Krylov subspace, or its basis given by $\mathbf{V}_{k+1}$.

To understand the convergence of the inexact process, we compare the norm of the computed residual for the inexact process, with perturbations of the form (7.10), to that of the exact method. We denote the computed residuals of both methods with, respectively, $\mathbf{r}_j$ and $\overline{\mathbf{r}}_j$. For the GMRES method these "residuals" are given by the following expressions:

$$\mathbf{r}_j^{\mathrm{G}} = \mathbf{V}_{j+1}(I - \underline{T}_j\underline{T}_j{}^{\dagger})e_1 \quad \text{and} \quad \overline{\mathbf{r}_j^{\mathrm{G}}} = \mathbf{V}_{j+1}(I - \overline{\underline{T}}_j\overline{\underline{T}}_j^{\dagger})e_1.$$

Since we have that $\overline{\underline{T}}_k = \underline{T}_k - \underline{E}_k$, we can apply standard perturbation theory for the least squares problem. For example, with Theorem 19.1 in [22] we can show that

$$\left|\|\overline{\mathbf{r}_k^{\mathrm{G}}}\|_2 - \|\mathbf{r}_k^{\mathrm{G}}\|_2\right| \leq \|\overline{\mathbf{r}_k^{\mathrm{G}}} - \mathbf{r}_k^{\mathrm{G}}\|_2 \leq (1 + 2\|\mathbf{A}\|_2\|\mathbf{A}^{-1}\|_2)\|\underline{E}_k\|_2.$$

This shows that if $\|\overline{\mathbf{r}_k^{\mathrm{G}}}\|_2 = \mathcal{O}(\varepsilon)$ all the η_j should be about ε in order to retain the speed of convergence of the exact method. This simple argument is not sufficient for explaining the fast convergence of the inexact method with the relaxation strategy (3.3), leaving some more work necessary. By generalizing Theorem 19.1 in [22], we get the following theorem.

THEOREM 7.1. *Let* $\mathbf{W}_k \equiv \widehat{\mathbf{A}}_k\mathbf{V}_k$ *and let* $\mathbf{P}_k$ *be the skew projection along* $\mathbf{V}_k$ *on* $\mathrm{span}(\mathbf{W}_k)$*:*

$$\mathbf{P}_k = \mathbf{W}_k(\mathbf{V}_k^*\mathbf{W}_k)^{-1}\mathbf{V}_k^* = \mathbf{V}_{k+1}\underline{\overline{T}}_k\overline{T}_k^{-1}\mathbf{V}_k^*.$$

Then, we have for the inexact FOM method

$$\|\overline{\mathbf{r}_k^{\mathrm{F}}} - \mathbf{r}_k^{\mathrm{F}}\|_2 \le \|\mathbf{I}-\mathbf{P}_k\|_2\|\mathbf{A}\|_2\|\underline{T}_k^\dagger\|_2\sum_{j=0}^{k-1}\eta_j(\|\mathbf{r}_j^{\mathrm{G}}\|_2 + \|\mathbf{r}_k^{\mathrm{F}}\|_2).$$

For the inexact GMRES method we have that

$$\|\overline{\mathbf{r}_k^{\mathrm{G}}} - \mathbf{r}_k^{\mathrm{G}}\|_2 \le \|\mathbf{A}\|_2\|\underline{T}_k^\dagger\|_2\sum_{j=0}^{k-1}\eta_j(\|\mathbf{r}_j^{\mathrm{G}}\|_2 + \|\mathbf{r}_k^{\mathrm{G}}\|_2).$$

Proof. We prove the first statement,

$$\begin{aligned}
\|\overline{\mathbf{r}_k^{\mathrm{F}}} - \mathbf{r}_k^{\mathrm{F}}\|_2 &= \|\underline{\overline{T}}_k\overline{T}_k^{-1}e_1 - \underline{T}_kT_k^{-1}e_1\|_2\\
&= \|[(\underline{\overline{T}}_k - \underline{T}_k) - \underline{\overline{T}}_k\overline{T}_k^{-1}(\overline{T}_k - T_k)]T_k^{-1}e_1\|_2\\
&= \|(\underline{E}_k - \underline{\overline{T}}_k\overline{T}_k^{-1}\underline{E}_k)T_k^{-1}e_1\|_2 = \|(\mathbf{I}-\mathbf{P}_k)\widetilde{\mathbf{F}}_kT_k^{-1}e_1\|_2\\
&\le \sum_{j=0}^{k-1}\|(\mathbf{I}-\mathbf{P}_k)\widetilde{\mathbf{f}}_j\|_2\,|e_{j+1}^*T_k^{-1}e_1|\\
&\le \|\mathbf{I}-\mathbf{P}_k\|_2\|\mathbf{A}\|_2\,\|\underline{T}_k^\dagger\|_2\sum_{j=0}^{k-1}\eta_j(\|\mathbf{r}_j^{\mathrm{G}}\|_2 + \|\mathbf{r}_k^{\mathrm{F}}\|_2),
\end{aligned}$$

where, in the last line, we have used Lemma 3.1. This proves the first statement.

For the proof for inexact GMRES, we define $\mathbf{Q}_k$ as the orthogonal projection onto $\mathrm{span}(\mathbf{W}_k)$; then

$$\mathbf{Q}_k = \mathbf{W}_k(\mathbf{W}_k^*\mathbf{W}_k)^{-1}\mathbf{W}_k^* = \mathbf{V}_{k+1}\underline{\overline{T}}_k\underline{\overline{T}}_k^\dagger\mathbf{V}_{k+1}^*.$$

We have that

$$\begin{aligned}
\|\overline{\mathbf{r}_k^{\mathrm{G}}} - \mathbf{r}_k^{\mathrm{G}}\|_2 &= \|\underline{\overline{T}}_k\underline{\overline{T}}_k^\dagger e_1 - \underline{T}_k\underline{T}_k^\dagger e_1\|_2\\
&= \|\underline{\overline{T}}_k\underline{\overline{T}}_k^\dagger(I - \underline{T}_k\underline{T}_k^\dagger)e_1 - (I - \underline{\overline{T}}_k\underline{\overline{T}}_k^\dagger)\underline{T}_k\underline{T}_k^\dagger e_1\|_2\\
&\le \|\underline{\overline{T}}_k\underline{\overline{T}}_k^\dagger(I - \underline{T}_k\underline{T}_k^\dagger)\|_2\,\|(I - \underline{T}_k\underline{T}_k^\dagger)e_1\|_2 + \|(I - \underline{\overline{T}}_k\underline{\overline{T}}_k^\dagger)\underline{E}_k\,\underline{T}_k^\dagger e_1\|_2\\
&\le \|(I - \underline{\overline{T}}_k\underline{\overline{T}}_k^\dagger)\underline{E}_k\,\underline{T}_k^\dagger\|_2\,\|\mathbf{r}_k^{\mathrm{G}}\|_2 + \|(I - \underline{\overline{T}}_k\underline{\overline{T}}_k^\dagger)\underline{E}_k\,\underline{T}_k^\dagger e_1\|_2\\
&= \|(\mathbf{I}-\mathbf{Q}_k)\widetilde{\mathbf{F}}_k\,\underline{T}_k^\dagger\|_2\,\|\mathbf{r}_k^{\mathrm{G}}\|_2 + \|(\mathbf{I}-\mathbf{Q}_k)\widetilde{\mathbf{F}}_k\,\underline{T}_k^\dagger e_1\|_2\\
&\le \|\widetilde{\mathbf{F}}_k\|_2\|\underline{T}_k^\dagger\|_2\,\|\mathbf{r}_k^{\mathrm{G}}\|_2 + \sum_{j=0}^{k}\|\widetilde{\mathbf{f}}_j\|_2\,|e_{j+1}^*\underline{T}_k^\dagger e_1|\\
&\le \|\mathbf{A}\|_2\,\|\underline{T}_k^\dagger\|_2\sum_{j=0}^{k-1}\eta_j(\|\mathbf{r}_j^{\mathrm{G}}\|_2 + \|\mathbf{r}_k^{\mathrm{G}}\|_2).
\end{aligned}$$

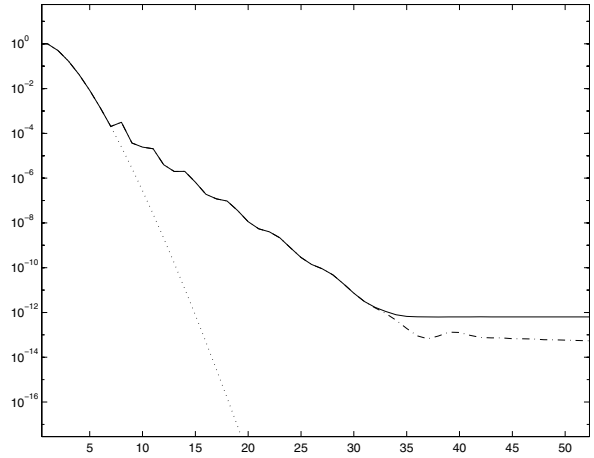


FIG. 7.1. *Convergence inexact FOM with* $\eta_j = \varepsilon = 10^{-12}$*: true residual* (—)*, computed residual* (- ·)*, and* $1/j!$ *(dotted) as a function of* j*.*

Here we used Lemma 3.1 and the identities $(I - \underline{\overline{T}}_k \underline{\overline{T}}_k^\dagger)\underline{T}_k = -(I - \underline{\overline{T}}_k \underline{\overline{T}}_k^\dagger)\underline{E}_k$ and $\|\underline{\overline{T}}_k \underline{\overline{T}}_k^\dagger (I - \underline{T}_k \underline{T}_k^\dagger)\|_2 = \|(I - \underline{\overline{T}}_k \underline{\overline{T}}_k^\dagger)\underline{T}_k \underline{T}_k^\dagger\|_2$, which, for example, can be found in [29]. □

This theorem shows that for special perturbations, the relaxation strategy also preserves the convergence speed of the exact method until the norm of the residuals becomes in the order of the required residual precision. Of course, this does not explain the often good results with relaxed GMRES that is observed, for example, in the experiments in [3]. However, this theorem is difficult to extend to more general perturbations since the Hessenberg reduction is not forward stable; see [32]. This means that small perturbations in the matrix-vector product can drastically change the resulting Hessenberg matrix. We emphasize that this does not necessarily imply a severe loss of convergence speed for general perturbations but only that the usefulness of the analytical approach taken here is limited. Nevertheless, small perturbations of the matrix-vector product can indeed delay convergence (but they seem not to have a big impact on the stagnation level). We illustrate this by the following experiment with inexact FOM. (Notice that the convergence of the computed residuals of inexact FOM and GMRES are related; see (7.3).)

The matrix $\mathbf{A} \in \mathbb{R}^{100\times 100}$ is lower bidiagonal with diagonal elements $(\mathbf{A})_{j,j} = j$ and has ones on its lower bidiagonal. For the right-hand side we have taken $\mathbf{b} = e_1$. It easily follows for this example that $\overline{T}_n = \mathbf{A}$ and the corresponding vector $\vec{\gamma}_j$ with $\vec{\gamma}_j^* \underline{T}_j = \vec{0}^*$ and $\vec{\gamma}_j^* e_1 = 1$ is given by $\gamma_j = (-1)^j j!$. Therefore we have that $\|\overline{\mathbf{r}^{\mathrm{F}}}_j\|_2 = 1/j!$. Figure 7.1 shows the convergence history of inexact FOM with $\eta_j = \varepsilon = 10^{-12}$. Although, the accuracy requirement is achieved (as expected), for the inexact method many more iterations are necessary to reach the required precision. An explanation is offered by the fact that the right-hand side is mainly oriented in the direction of a few eigenvectors of $\mathbf{A}$ and the errors in the matrix-vector product introduce components in directions for which convergence is slow. We mention that convergence of GMRES for this system for general right-hand sides is much slower than for the right-hand side taken in this example. We must, however, emphasize that this example is academic since also in finite precision computations the convergence can be much slower than the exact expression for which the residuals suggests.

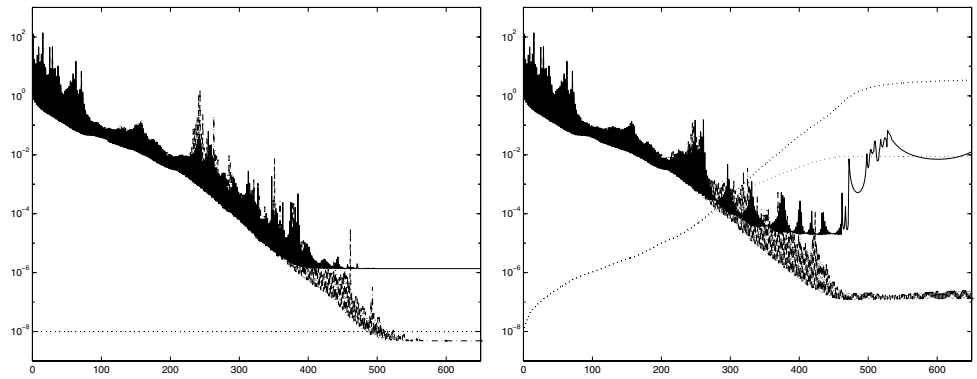


FIG. 8.1. *True residuals CG (solid), Orthores (-·), Rutishauser's variant (dots), η_j (dotted) as a function of j. In both pictures $\varepsilon = 10^{-8}$. Left: $\eta_j = \varepsilon$. Right: $\eta_j = \varepsilon/\rho_j$.*

8. Numerical experiments. In this section we conduct an experiment with inexact CG and its variants from section 6. For experiments with inexact GMRES we refer the reader to [3]. All experiments are done in Matlab.

The linear system comes from the computation of quark propagators using Wilson fermions in QCD. The matrix $\mathbf{D}_W$ is `CONF6.0-0.0014x4.2000` from the Matrix Market. This matrix is complex valued and contains 3072 unknowns. The matrix has the following property, e.g., [8], $\mathbf{\Gamma_5}\,\mathbf{D}_W = \mathbf{D}_W^*\,\mathbf{\Gamma_5}$ with $\mathbf{\Gamma_5} \equiv \mathbf{I} \otimes (\gamma_5 \otimes \mathbf{I}_3)$ and

$$\gamma_5 \equiv \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

The Hermitian matrix $\mathbf{A}$ is now given by $\mathbf{A} = \mathbf{\Gamma_5}\,\mathbf{D}_W$. This matrix is highly indefinite. For the right-hand side we have taken a complex random vector of unit length. To simulate an inexact matrix-vector product we have added in step j of CG, a random complex vector. We have not taken into account the norm of $\mathbf{A}$ in our experiments.

Figure 8.1 shows the results for inexact CG, Orthores, and Rutishauser's variant when a residual precision of $\mathcal{O}(\varepsilon)$ is required with $\varepsilon = 10^{-8}$. The left picture shows the results for a constant precision ($\eta_j = \varepsilon$) and the right picture for the relaxation strategy from Section 6.4 ($\eta_j = \varepsilon/\rho_j$).

For $\eta_j = 10^{-8}$ we see that the three-term recurrence is superior to the coupled two-term recurrence. This can be explained by our analysis and the large residuals in the initial steps. This advantage remains if we apply the relaxation strategy from section 6.4 (although we lose some additional digits compared to the constant precision case).

9. Conclusions and outlook. In this paper we have investigated the effect of approximately computed matrix-vector products on the convergence and accuracy of various Krylov subspace methods. This analysis was used to derive suitable relaxation strategies for these methods. Our results provide insights into the mechanisms behind the successful results with the relaxation strategies of Bouras and Frayssé in [3] and Bouras, Frayssé, and Giraud in [4]. Furthermore, it was shown that for the CG method the three-term recurrence can offer advantages over the standard coupled two-term

recurrence in case the matrix is indefinite and suffers from large intermediate residuals or peaks in the convergence curve. This was illustrated in section 8.

For methods like Richardson iteration and Chebyshev iteration it is necessary that the residuals are computed in an incremental matter in order for a relaxation strategy to be possible. We illustrated, by the example of CG versus Orthores for indefinite problems, that it is the underlying way the Krylov subspace is constructed that is of importance. By comparing inexact FOM and inexact GMRES we saw that the optimality properties of the residuals are not of influence on the attainable accuracy in the end. Therefore, a relaxation strategy for GMRES should also work for FOM, since the Krylov subspace is constructed in the same matter, i.e., using inexact Arnoldi.

Studying the convergence of the inexact methods is a more difficult problem. Stationary methods construct residual polynomials that are small everywhere on a predefined interval. For these types of methods we could prove that, with our relaxation strategies, convergence is as fast for the exact method. For GMRES and CG this is a much more difficult problem. For the GMRES method we have given some results in case the perturbations are of a special form. In future work we plan to further study the effect of inexact matrix-vector products on optimal Krylov subspace methods. And, in particular, the effect of increasing the error during the process.

As a side product of our work, we have shown that using the matrix formulations of the Krylov subspace methods in some cases can simplify the analysis of the residual gap, which is a problem that frequently occurs in analyses of the attainable accuracy of subspace methods. In particular, for three-term recurrences insightful expressions can be easily obtained for the likes of Chebyshev method and Orthores.

In future work we want to apply the observations in this paper to the simulation of overlap fermions (as mentioned in the beginning of section 3) and combine this with the work in [30] for the computation of the matrix sign function acting on a vector. Furthermore, we plan to extend the analysis in this paper to a rounding error analysis for the different variants of CG for indefinite Hermitian systems (and the BiCG method) in order to understand the effect of the different types of breakdown on the residual gap.

Postscript. After the submission of this paper, the presentation in [25] of Simoncini and Szyld resulted in the paper [26]. We discuss some differences with this work. The analysis in the presentation [25] mainly focused on the inexact GMRES and inexact FOM method and is based on showing that the true residuals satisfy a *quasi-orthogonality condition* of the form $\|\mathbf{U}_k^*(\mathbf{b} - \mathbf{A}\mathbf{x}_k)\| \leq \mathcal{O}(\varepsilon)$ for some matrix $\mathbf{U}_k$. It is interesting to notice that the quasi-orthogonality is equal to a projection of the residual gap. Therefore, in their presentation, the authors in the end presented a result similar to our Lemma 3.1 to bound this quasi-orthogonality. Paper [26] considers a large number of practical applications. Moreover, the approach taken in the analysis is very different. In this paper, we are interested in the convergence and stagnation level of the true residuals which are indicators for the quality of the iterates. The basis of our analysis is the splitting into a study of the residual gap, which is connected to the stagnation level, and the convergence and stagnation of the computed residuals. In [26], the authors consider two aspects of inexact Krylov subspace methods: the already mentioned quasi-orthogonality of the true residuals and the variational properties of inexact GMRES and inexact FOM method. (This is equivalent to the observation in Section 7 that the computed residuals in inexact GMRES and FOM are residuals of an exact GMRES/FOM process applied to a "nearby"

matrix.) There seems to be no discussion in [26] about the direct consequence of quasi-orthogonality and the conserved variational properties of the Krylov subspace method on the stagnation level and convergence speed of the inexact method.

Acknowledgments. The authors are thankful to Valeria Simoncini for providing them with a copy of the slides from [25]. We are thankful to the referees for their constructive comments. Their remarks have helped us to improve the presentation of this paper.

Appendix A. A technical result.

LEMMA A.1. *Let Δ_k be the k by k matrix with zeros entries everywhere except at the positions $(j-1,j)$ and $(j,j-1)$, where it has the value one and the $(2,1)$ element is 2. Then*

$$|e_j^*\Delta_k^{2i}e_j| \le 2\frac{(2i)!}{(i!)^2} \quad \textit{for all } i,j \ge 1,\ j \le k.$$

Proof. Let $\mathbb{R}^{\mathbb{N}}$ and $\mathbb{R}^{\mathbb{Z}}$ be the space of vectors with indices in $\mathbb{N}$ and $\mathbb{Z}$, respectively. Consider the map $\widetilde{\Delta}$ on $\mathbb{R}^{\mathbb{Z}}$ given by $\widetilde{\Delta}e_j \equiv e_{j-1}+e_{j+1}$ for all $j \in \mathbb{Z}$. Extend the map Δ_k on $\mathbb{R}^k$ to the map Δ on $\mathbb{R}^{\mathbb{N}}$ given by $\Delta e_j \equiv e_{j-1}+e_{j+1}$ for $j>1$ and $\Delta e_1 \equiv 2e_2$. Note that $0 \le e_i^*\Delta_k e_j \le e_i^*\Delta e_j$ for all $i,j \in \mathbb{N}$: here we follow the convention that $\Delta_k e_j = \mathbf{0}$ if $j > k$. Consider the linear map $\mathbf{P} : \mathbb{R}^{\mathbb{Z}} \to \mathbb{R}^{\mathbb{N}}$ defined by $\mathbf{P}e_{j+1} = e_{|j|+1}$. One can easily check that $\mathbf{P}\widetilde{\Delta}e_j = \Delta\,\mathbf{P}e_j$ for all $j \in \mathbb{Z}$. Therefore, $\mathbf{P}\widetilde{\Delta} = \Delta\,\mathbf{P}$, and for $j \ge 0$, we have that

$$\Delta^{2i}e_{j+1} = \Delta^{2i}\mathbf{P}e_{j+1} = \mathbf{P}(\widetilde{\Delta}^{2i}e_{j+1}) = \mathbf{P}\left(\sum_{\ell=0}^{2i}\frac{(2i)!}{\ell!(2i-\ell)!}e_{j-2i+2\ell+1}\right).$$

If $i<j$ then $|j-2i+2\ell|+1 = j+1$ only if $\ell = i$. Hence, if $i<j$ we find that $e_{j+1}^*\Delta_k^{2i}e_{j+1} \le e_{j+1}^*\Delta^{2i}e_{j+1} = \frac{2i!}{(i!)^2}$. If $\ell \equiv i-j \ge 0$ then $|j-2i+2\ell|+1 = j+1$ and $e_{j+1}^*\Delta_k^{2i}e_{j+1} \le e_{j+1}^*\Delta^{2i}e_{j+1} = \frac{2i!}{(i!)^2} + \frac{2i!}{(i-j)!(i+j)!} \le 2\frac{2i!}{(i!)^2}$. ∎

REFERENCES

[1] R. E. BANK AND T. F. CHAN, *An analysis of the composite step biconjugate gradient method*, Numer. Math., 66 (1993), pp. 295–319.

[2] Å. BJÖRCK, T. ELFVING, AND Z. STRAKOŠ, *Stability of conjugate gradient and Lanczos methods for linear least squares problems*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 720–736.

[3] A. BOURAS AND V. FRAYSSÉ, *A Relaxation Strategy for Inexact Matrix-Vector Products for Krylov Methods*, Technical Report TR/PA/00/15, CERFACS, France, 2000.

[4] 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.

[5] P. N. BROWN, *A theoretical comparison of the Arnoldi and GMRES algorithms*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 58–78.

[6] B. FISCHER, *Polynomial Based Iteration Methods for Symmetric Linear Systems*, John Wiley & Sons Ltd., Chichester, 1996.

[7] L. FOX AND I. B. PARKER, *Chebyshev Polynomials in Numerical Analysis*, Oxford University Press, London, 1972.

[8] A. FROMMER, T. LIPPERT, B. MEDEKE, AND K. SCHILLING, EDS., *Numerical Challenges in Lattice Quantum Chromodynamics*, Lecture Notes in Computational Science and Engineering, Springer Verlag, Heidelberg, 2000.

[9] E. GILADI, G. H. GOLUB, AND J. B. KELLER, *Inner and outer iterations for the Chebyshev algorithm*, SIAM J. Numer. Anal., 35 (1998), pp. 300–319.

[10] G. H. Golub and M. L. Overton, *The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems*, Numer. Math., 53 (1988), pp. 571–593.
[11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The John Hopkins University Press, Baltimore, London, 3rd ed., 1996.
[12] G. H. Golub and Q. Ye, *Inexact preconditioned conjugate gradient method with inner-outer iteration*, SIAM J. Sci. Comput., 21 (1999), pp. 1305–1320.
[13] G. H. Golub, Z. Zhang, and H. Zha, *Large sparse symmetric eigenvalue problems with homogeneous linear constraints: the Lanczos process with inner-outer iterations*, Linear Algebra Appl., 309 (2000), pp. 289–306.
[14] A. Greenbaum, *Behavior of slightly perturbed Lanczos and conjugate-gradient recurrences*, Linear Algebra Appl., 113 (1989), pp. 7–63.
[15] A. Greenbaum, *Estimating the attainable accuracy of recursively computed residual methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 535–551.
[16] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Frontiers Appl. Math. 17, SIAM, Philadelphia, 1997.
[17] A. Greenbaum, V. L. Druskin, and L. A. Knizhnerman, *On solving indefinite symmetric linear systems by means of the Lanczos method*, Zh. Vychisl. Mat. Mat. Fiz., 39 (1999), pp. 371–377.
[18] M. H. Gutknecht, *Lanczos-type solvers for nonsymmetric linear systems of equations*, in Acta Numerica, 1997, Cambridge Univ. Press, Cambridge, UK, 1997, pp. 271–397.
[19] M. H. Gutknecht and S. Röllin, *The Chebyshev iteration revisited*, Parallel Computing, 28 (2002), pp. 263–283.
[20] M. H. Gutknecht and Z. Strakoš, *Accuracy of two three-term and three two-term recurrences for Krylov space solvers*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 213–229.
[21] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436.
[22] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
[23] C. C. Paige, *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix*, J. Inst. Math. Appl., 18 (1976), pp. 341–349.
[24] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.
[25] V. Simoncini and D. B. Szyld, *Flexible inner-outer Krylov methods (and inexact Krylov methods)*, presentation, Zürich, 2002.
[26] 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.
[27] G. L. G. Sleijpen, H. A. van der Vorst, and D. R. Fokkema, *BiCGstab(ℓ) and other hybrid Bi-CG methods*, Numer. Algorithms, 7 (1994), pp. 75–109.
[28] G. L. G. Sleijpen, H. A. van der Vorst, and J. Modersitzki, *Differences in the effects of rounding errors in Krylov solvers for symmetric indefinite linear systems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 726–751.
[29] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, San Diego, 1990.
[30] J. van den Eshof, A. Frommer, T. Lippert, K. Schilling, and H. van de Vorst, *Numerical methods for the QCD overlap operator:* I. *sign-function and error bounds*, Comput. Phys. Comm., 146 (2002), pp. 203–224.
[31] H. A. van der Vorst and C. Vuik, *GMRESR: a family of nested GMRES methods*, Numer. Linear Algebra Appl., 1 (1994), pp. 369–386.
[32] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.
[33] H. Woźniakowski, *Numerical stability of the Chebyshev method for the solution of large linear systems*, Numer. Math., 28 (1977), pp. 191–209.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 1, pp. 154–169

STABILITY ESTIMATES ON THE JACOBI AND UNITARY HESSENBERG INVERSE EIGENVALUE PROBLEMS*

LEONID KNIZHNERMAN†

Abstract. Perturbation bounds for the Jacobi inverse eigenvalue problem (JIEP), which are more realistic than the earlier ones, are proved and illustrated by numerical experiments. The technique of orthonormal polynomials and integral representation of Hankel determinants is used. The same technique is then applied to the unitary Hessenberg inverse eigenvalue problem (UHIEP).

Key words. Jacobi inverse eigenvalue problem, unitary Hessenberg inverse eigenvalue problem, stability estimates, Lanczos method, Gaussian quadrature formula, orthonormal polynomials, Szegö polynomials

AMS subject classification. 65F18

DOI. 10.1137/S0895479802410098

1. Introduction. Hochstadt [15] stated the following problem, called the Jacobi inverse eigenvalue problem (JIEP): given $n \geq 1$ and real numbers $\lambda_1 < \mu_1 < \lambda_2 < \cdots < \mu_{n-1} < \lambda_n$, find a Jacobi matrix

$$J = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-2} & \alpha_{n-2} & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{pmatrix}, \qquad \alpha_k, \beta_k \in \mathbf{R}, \qquad \beta_k > 0, \tag{1.1}$$

such that λ_k are the eigenvalues of J and μ_k are the eigenvalues of its principal lower $(n-1) \times (n-1)$ submatrix. This problem, including its practical importance, is discussed in [5, sect. 4].

The related stability problem (to estimate the perturbation of J in terms of the spectrum's perturbation) was considered in [13, 22], where perturbation estimates with respectively nonconstructive[1] and constructive coefficients were proved. Though the result of [22] is undoubtedly of principal importance, its coefficient may be huge (probably because of straightforward work with the power basis of a Krylov subspace).

In this paper we shall try to give perturbation bounds that are not too far from reality. The technique of orthogonal polynomials and integral representation of Hankel determinants[2] will be exploited. We shall first consider a similar problem [9] in terms of the associated Gaussian quadrature formula [12] and the underlying Lanczos process [19, Chap. 13]; the fact that this intermediate problem is well conditioned was conjectured in [10, sect. 3.1]. Then we shall reduce investigation of perturbation of μ_i to that of weights of the quadrature formula. The results of numerical experiments will be shown.

*Received by the editors June 24, 2002; accepted for publication (in revised form) by I. S. Dhillon November 4, 2003; published electronically August 27, 2004.
http://www.siam.org/journals/simax/26-1/41009.html

†Central Geophysical Expedition, Narodnogo Opolcheniya Street, House 38, Building 3, Moscow 123298, Russia (mmd@cge.ru).

[1]This means that only the existence theorem was proved.

[2]Earlier Hankel determinants were used in an algorithm of computing singular values [21]; Hankel matrices played an important role in the analysis of reduction to tridiagonal form [18].

Finally, we shall briefly present analogous results for the unitary Hessenberg inverse eigenvalue problem (UHIEP).

2. Preliminaries. We shall use the following notation: μ is a positive measure on $\mathbf{R}$, Q_k ($k \in \mathbf{N}$) are the corresponding orthonormal polynomials, $s_k = \int x^k \, d\mu(x)$ are its moments, α_k and β_k are the coefficients of the Lanczos recurrence

$$xQ_k(x) = \beta_{k+1}Q_{k+1}(x) + \alpha_{k+1}Q_k(x) + \beta_k Q_{k-1}(x) \qquad (k \in \mathbf{N}), \tag{2.1}$$
$$Q_{-1} \equiv 0, \qquad Q_0 = 1/\sqrt{s_0}$$

(Q_k can be considered as the Lanczos vectors of the Lanczos process in $L_{2,\mu}$ with the initial vector 1 (a constant function) and the operator of multiplication by x),

$$H_k = \begin{vmatrix} s_0 & s_1 & \dots & s_{k-1} \\ s_1 & s_2 & \dots & s_k \\ \vdots & \vdots & \dots & \vdots \\ s_{k-1} & s_k & \dots & s_{2k-2} \end{vmatrix} > 0 \quad \text{and} \quad G_k = \begin{vmatrix} s_1 & s_2 & \dots & s_k \\ s_2 & s_3 & \dots & s_{k+1} \\ \vdots & \vdots & \dots & \vdots \\ s_k & s_{k+1} & \dots & s_{2k-1} \end{vmatrix} \tag{2.2}$$

are Hankel determinants.

The formulae

$$Q_k(x) = \frac{1}{\sqrt{H_k H_{k+1}}} \begin{vmatrix} s_0 & s_1 & \dots & s_k \\ s_1 & s_2 & \dots & s_{k+1} \\ \vdots & \vdots & \dots & \vdots \\ s_{k-1} & s_k & \dots & s_{2k-1} \\ 1 & x^1 & \dots & x^k \end{vmatrix}, \tag{2.3}$$

$$H_k = \frac{1}{k!} \int \cdots \int \prod_{1 \le i < j \le k} (t_j - t_i)^2 \, d\mu(t_1) \dots d\mu(t_k), \tag{2.4}$$

and

$$\text{leading coefficient of } Q_k = \sqrt{\frac{H_k}{H_{k+1}}} = \frac{1}{\sqrt{s_0}\,\beta_1 \cdots \beta_k} \tag{2.5}$$

(see [17, Chap. 2, sect. 5] or [20, Chap. 2]) will be used.

We shall assume from now on that the measure μ has a finite set of increase points λ_i (spectrum), $\lambda_1 < \cdots < \lambda_n$, with weights $\omega_i = \mu(\lambda_i) > 0$. Note that the Lanczos coefficients α_k and β_k are the same as in (1.1), provided the weights are related to μ_i by the formula

$$\omega_i = \prod_{j=1}^{n-1} (\lambda_i - \mu_j) \Big/ \prod_{1 \le j \le n,\ j \ne i} (\lambda_i - \lambda_j). \tag{2.6}$$

In this (discrete) situation the polynomials Q_k are defined for $0 \le k \le n-1$, the quantities H_k and G_k are defined for $0 \le k \le n$ and $\beta_n = 0$.

Let

$$d = \min_{1 \le i,j \le n,\ i \ne j} |\lambda_i - \lambda_j| > 0 \quad \text{and} \quad d_2 = \min_{1 \le i \le n,\ 1 \le j \le n-1} |\lambda_i - \mu_j| > 0$$

be the separations.

In the case $\lambda_1 > 0$ we shall also consider the Stieltjes continued fraction representation of the impedance function (useful in cutting computational domains when numerically solving differential equations; see, e.g., [6])

$$\sum_{i=1}^{n} \frac{\omega_i}{x+\lambda_i} = \cfrac{1}{\widehat{h}_1 x + \cfrac{1}{h_1 + \cfrac{1}{\widehat{h}_2 x + \ldots \cfrac{1}{h_{n-1} + \cfrac{1}{\widehat{h}_n x + \cfrac{1}{h_n}}}}}}. \tag{2.7}$$

According to a theorem by Stieltjes [8, Chap. 16, Thm. 15], the fraction (2.7) has real positive parameters h_k and $\widehat{h}_k$. They are determined by the formulae [16, (13.17)]

$$h_k = \frac{H_k^2}{G_{k-1}G_k}, \qquad \widehat{h}_k = \frac{G_{k-1}^2}{H_{k-1}H_k}. \tag{2.8}$$

The symbol $\mathtt{RP}$ denotes relative perturbation: $\mathtt{RP}\, a = |\widetilde{a}/a - 1|$, where $\widetilde{a}$ is a perturbed value of a quantity $a \neq 0$. Analogously, the symbol $\mathtt{AP}$ denotes absolute perturbation: $\mathtt{AP}\, a = |\widetilde{a} - a|$.

3. Auxiliary assertions. To shorten further formulae, we introduce the family of functions[3]

$$\overset{m}{\mathcal{D}}(\delta) = \frac{\delta}{1-m\delta}, \qquad 0 \le \delta < \frac{1}{m}, \qquad m \ge 1, \qquad \mathcal{D} = \overset{1}{\mathcal{D}}. \tag{3.1}$$

These monotonically increasing functions possess the following simple properties:

$$\overset{m}{\mathcal{D}}(\delta) = \delta + O(\delta^2) \quad \text{as} \quad \delta \to +0;$$

$$\mathcal{D}\left[\overset{m}{\mathcal{D}}(\delta)\right] = \overset{m+1}{\mathcal{D}}(\delta), \qquad 0 \le \delta < \frac{1}{m+1};$$

$$\overset{m}{\mathcal{D}}(\delta) \le \overset{m+1}{\mathcal{D}}(\delta), \qquad 0 \le \delta < \frac{1}{m+1};$$

$$\overset{m}{\mathcal{D}}(\delta_1) + \overset{m}{\mathcal{D}}(\delta_2) \le \overset{m}{\mathcal{D}}(\delta_1+\delta_2), \qquad \delta_1 \ge 0, \quad \delta_2 \ge 0, \quad \delta_1+\delta_2 \le \frac{1}{m}.$$

LEMMA 3.1. *Let $k \in \mathbf{N}$ and ε_i $(1 \le i \le k)$ be real numbers such that*

$$\sum_{i=1}^{k} |\varepsilon_i| < 1. \tag{3.2}$$

Then

$$\left| \prod_{i=1}^{k} (1+\varepsilon_i) - 1 \right| \le \mathcal{D}\left(\sum_{i=1}^{k} |\varepsilon_i| \right). \tag{3.3}$$

[3]This is a referee's suggestion; it will enable us to avoid taking care of $O(\delta^2)$ terms (cf. [14]).

Proof. We shall use the inequalities

$$\frac{x}{1+x} \le \log(1+x) \le x \qquad (x > -1),$$
$$1 + x \le \exp(x) \le \frac{1}{1-x} \qquad (x < 1)$$

[1, items 4.1.33, 4.2.30–31]. We obtain with the use of (3.2)

$$\prod_{i=1}^{k}(1+\varepsilon_i) - 1 = \exp\left[\sum_{i=1}^{k}\log(1+\varepsilon_i)\right] - 1 \le \exp\left(\sum_{i=1}^{k}\varepsilon_i\right) - 1 \le \frac{1}{1-\sum_{i=1}^{k}\varepsilon_i} - 1 = \frac{\sum_{i=1}^{k}\varepsilon_i}{1-\sum_{i=1}^{k}\varepsilon_i} \le \frac{\sum_{i=1}^{k}|\varepsilon_i|}{1-\sum_{i=1}^{k}|\varepsilon_i|} \tag{3.4}$$

and

$$\prod_{i=1}^{k}(1+\varepsilon_i) - 1 = \exp\left[\sum_{i=1}^{k}\log(1+\varepsilon_i)\right] - 1 \ge \exp\left(\sum_{i=1}^{k}\frac{\varepsilon_i}{1+\varepsilon_i}\right) - 1 \ge \sum_{i=1}^{k}\frac{\varepsilon_i}{1+\varepsilon_i} \ge -\sum_{i=1}^{k}\frac{|\varepsilon_i|}{1-|\varepsilon_i|} \ge -\frac{\sum_{i=1}^{k}|\varepsilon_i|}{1-\sum_{i=1}^{k}|\varepsilon_i|}. \tag{3.5}$$

The inequalities (3.4) and (3.5) imply (3.3). □

Lemma 3.1 will help us to estimate relative perturbations, because its assertion (3.3) can be reformulated as

$$\mathtt{RP}\prod_{i=1}^{k} a_i \le \mathcal{D}\left(\sum_{i=1}^{k}\mathtt{RP}\, a_i\right). \tag{3.6}$$

Note that if $\mathtt{RP}\, a \le \mathcal{D}(\delta_1)$ and $\mathtt{RP}\, b \le \mathcal{D}(\delta_2)$, then

$$\mathtt{RP}(ab) \le \mathcal{D}[\mathcal{D}(\delta_1) + \mathcal{D}(\delta_2)] \le \overset{2}{\mathcal{D}}(\delta_1 + \delta_2) \tag{3.7}$$

and

$$\mathtt{RP}\,\frac{1}{a} = \left|\frac{\frac{1}{\mathfrak{a}} - \frac{1}{a}}{\frac{1}{a}}\right| = \left|\frac{\frac{\mathfrak{a}-a}{a}}{\frac{\mathfrak{a}-a}{a}+1}\right| \le \frac{\mathtt{RP}\, a}{1-\mathtt{RP}\, a} = \mathcal{D}(\mathtt{RP}\, a) = \overset{2}{\mathcal{D}}(\delta_1). \tag{3.8}$$

The next two lemmas evaluate perturbation of Hankel determinants (2.2).

Lemma 3.2. *Let* $\max_{1\le i\le n}\mathtt{AP}\,\lambda_i \le \varepsilon$ *and* ω_i *remain unchanged. Then we have*[4]

$$\mathtt{RP}\, H_k \le \mathcal{D}\left(\frac{4\varepsilon k\log k}{d}\right). \tag{3.9}$$

Proof. Due to the discreteness of μ we can rewrite formula (2.4) as

$$H_k = \sum_{t_1<\cdots<t_k,\ t_1,\ldots,t_k\in\{\lambda_1,\ldots,\lambda_n\}}\ \prod_{1\le i<j\le k}(t_j - t_i)^2 \prod_{i=1}^{k}\mu(t_i).$$

[4]Provided the value of $\mathcal{D}$ is well defined according to (3.1).

We now estimate $\mathtt{RP}\prod_{1\le i<j\le k}(t_j-t_i)$ for a k-tuple $(t_1,\dots,t_k)$. Noting that for $k-1$ index pairs (i,j) $t_j-t_i\ge d$, for $k-2$ other index pairs $t_j-t_i\ge 2d$, … , for 1 remaining index pair $t_j-t_i\ge(k-1)d$, we derive

$$\sum_{1\le i<j\le k}\mathtt{RP}(t_j-t_i)\le\frac{2\varepsilon}{d}\sum_{i=1}^{k-1}\frac{k-i}{i}=\frac{2\varepsilon}{d}\left[k\sum_{i=1}^{k-1}\frac{1}{i}-(k-1)\right]\le\frac{2\varepsilon k\log k}{d}, \tag{3.10}$$

which gives (3.9) by virtue of (3.6). □

Estimate (3.9) and further similar estimates are valid when the quantities under the $\overset{m}{\mathcal{D}}$ symbols obey the inequality in (3.1).

LEMMA 3.3. *Suppose that* $\lambda_1>0$. *Let* $\max_{1\le i\le n}\mathtt{AP}\,\lambda_i\le\varepsilon$ *and* ω_i *remain unchanged. Then the estimate*

$$\mathtt{RP}\,G_k\le\mathcal{D}\left(\frac{4\varepsilon k\log k}{d}+\frac{\varepsilon k}{\lambda_1}\right) \tag{3.11}$$

holds.

Proof. Because of the positivity of the spectrum we can define a positive measure ν by the formula $d\nu(\lambda)=\lambda\,d\mu(\lambda)$. Note that G_k are the analogues of the quantities H_k for the measure ν. The measure ν has the same spectrum as μ, but each weight ω_i is changed to $\lambda_i\omega_i$, so we have

$$G_k=\sum_{t_1<\dots<t_k,\ t_1,\dots,t_k\in\{\lambda_1,\dots,\lambda_n\}}\ \prod_{1\le i<j\le k}(t_j-t_i)^2\prod_{i=1}^{k}(t_i\omega_i).$$

Recalling (3.10) and additionally noting that

$$\sum_{i=1}^{k}\mathtt{RP}\,t_i\le\sum_{i=1}^{k}\frac{\varepsilon}{\lambda_i}\le\frac{\varepsilon k}{\lambda_1},$$

we obtain (3.11). □

4. Influence of nodes' perturbation.

THEOREM 4.1. *Under the condition of Lemma* 3.2 *the bound*

$$\mathtt{RP}\,\beta_k\le\frac{1}{2}\overset{3}{\mathcal{D}}\left[\frac{16\varepsilon k\log(k+1)}{d}\right] \tag{4.1}$$

takes place.

Proof. It follows from formula (2.5) by induction that

$$\beta_k=\frac{\sqrt{H_{k-1}H_{k+1}}}{H_k} \tag{4.2}$$

(see [11]). Formulae (3.9) and (4.2) in view of (3.6)–(3.8) imply

$$\begin{aligned}\mathtt{RP}\,\beta_k^2&\le\mathcal{D}\left(\mathtt{RP}\,H_{k-1}+\mathtt{RP}\,H_{k+1}+2\,\mathtt{RP}\,H_k^{-1}\right)\\&\le\mathcal{D}\left\{\mathcal{D}\left[\frac{4\varepsilon(k-1)\log(k-1)}{d}\right]+\mathcal{D}\left[\frac{4\varepsilon(k+1)\log(k+1)}{d}\right]+2\overset{2}{\mathcal{D}}\left(\frac{4\varepsilon k\log k}{d}\right)\right\}\\&\le\mathcal{D}\left\{\overset{2}{\mathcal{D}}\left[\frac{8\varepsilon k\log(k+1)}{d}\right]+2\overset{2}{\mathcal{D}}\left(\frac{4\varepsilon k\log k}{d}\right)\right\}\\&\le\overset{3}{\mathcal{D}}\left[\frac{16\varepsilon k\log(k+1)}{d}\right].\end{aligned}$$

□

THEOREM 4.2. *Under the condition of Lemma* 3.3 *the bound*

$$\mathtt{RP}\,\alpha_k \le \overset{3}{\mathcal{D}}\left(\frac{16\varepsilon k\log k}{d}+\frac{2\varepsilon k}{\lambda_1}\right) \tag{4.3}$$

is valid.

Proof. Substituting $x=0$ into the determinant formula (2.3) and decomposing the determinant in the last row, we obtain

$$Q_k(0)=\frac{(-1)^k G_k}{\sqrt{H_k H_{k+1}}}.$$

Now from the partial case of the recurrence (2.1) at $x=0$ and from (4.2) derive

$$\begin{aligned}\alpha_k &= -\frac{\beta_{k-1}Q_{k-2}(0)+\beta_k Q_k(0)}{Q_{k-1}(0)} \qquad (4.4)\\ &= \left(\frac{\sqrt{H_{k-2}H_k}}{H_{k-1}}\cdot\frac{G_{k-2}}{\sqrt{H_{k-2}H_{k-1}}}+\frac{\sqrt{H_{k-1}H_{k+1}}}{H_k}\cdot\frac{G_k}{\sqrt{H_k H_{k+1}}}\right)\frac{\sqrt{H_{k-1}H_k}}{G_{k-1}}\\ &= \frac{H_k G_{k-2}}{H_{k-1}G_{k-1}}+\frac{H_{k-1}G_k}{H_k G_{k-1}}.\end{aligned}$$

(Recall that $\beta_n=0$, so some terms disappear when $k=n$.) Formulae (3.9) and (3.11) by virtue of (3.6)–(3.8) give the bound

$$\begin{aligned}\mathtt{RP}\,\frac{H_k G_{k-2}}{H_{k-1}G_{k-1}} &\le \mathcal{D}\left(\mathtt{RP}\,H_k+\mathtt{RP}\,G_{k-2}+\mathtt{RP}\,H_{k-1}^{-1}+\mathtt{RP}\,G_{k-1}^{-1}\right)\\ &\le \mathcal{D}\left[\mathcal{D}\left(\frac{4\varepsilon k\log k}{d}\right)+\mathcal{D}\left(\frac{4\varepsilon k\log k}{d}+\frac{\varepsilon k}{\lambda_1}\right)\right.\\ &\quad \left.+\overset{2}{\mathcal{D}}\left(\frac{4\varepsilon k\log k}{d}\right)+\overset{2}{\mathcal{D}}\left(\frac{4\varepsilon k\log k}{d}+\frac{\varepsilon k}{\lambda_1}\right)\right]\\ &\le \overset{3}{\mathcal{D}}\left(\frac{16\varepsilon k\log k}{d}+\frac{2\varepsilon k}{\lambda_1}\right)\end{aligned}$$

and the analogous bound for $\frac{H_{k-1}G_k}{H_k G_{k-1}}$ that follow (4.3). □

Remark 1. In the case of $\lambda_1\le 0$ an additive shift can enable one to obtain estimates for $\mathtt{AP}\,\alpha_k$. Indeed, put $s=d-\lambda_1>0$. Applying Theorem 4.2 to the exact eigenvalues λ_i+s and perturbed eigenvalues $\widetilde{\lambda}_i+s$ (and taking into account that the new α_k, $\widetilde{\alpha}_k$ equal α_k+s, $\widetilde{\alpha}_k+s$, respectively, and that the new λ_1 equals d), we deduce

$$\frac{|\widetilde{\alpha}_k-\alpha_k|}{\alpha_k-\lambda_1+d}\le\overset{3}{\mathcal{D}}\left[\frac{\varepsilon k(16\log k+2)}{d}\right],$$

whence

$$\begin{aligned}|\widetilde{\alpha}_k-\alpha_k| &\le \overset{3}{\mathcal{D}}\left[\frac{\varepsilon k(16\log k+2)}{d}\right](\alpha_k-\lambda_1+d)\\ &\le \overset{3}{\mathcal{D}}\left[\frac{\varepsilon k(16\log k+2)}{d}\right](\lambda_n-\lambda_1+d).\end{aligned}$$

THEOREM 4.3. *Under the condition of Lemma* 3.3 *for the parameters of continued fraction* (2.7) *the estimates*

$$\mathtt{RP}\, h_k \ \textit{and}\ \mathtt{RP}\, \widehat{h}_k \le \overset{3}{\mathcal{D}} \left(\frac{16\varepsilon k \log k}{d} + \frac{2\varepsilon k}{\lambda_1} \right)$$

take place.

Proof. We use representations (2.8) and estimates (3.9) and (3.11). □

5. Numerical experiments. Numerical experiments were carried out on a PC in double precision. The Lanczos algorithm with full reorthogonalization[5] was implemented to solve the inverse spectral problem, as recommended in [5, sect. 4.5]. The perturbations of the eigenvalues were chosen in the ε-vicinities of the exact eigenvalues with $\varepsilon = 10^{-10}$ for Examples 1 and 2 and $\varepsilon = 10^{-5}$ for Example 3 in order to generate maximal possible errors, allowing for the weights to remain the same.

It is known [9, sect. 3.1] that the recurrence coefficients are less sensitive to the weights' perturbation than that of the nodes, so in our numerical tests we perturbed only the latter.

Two of the examples were taken (up to an additive shift) from [9, sect. 4].

Example 1 (see Figure 1). Equidistant spectrum with equal weights: $n = 100$, $\lambda_i = i/n,\ \omega_i = 1/n$.

Example 2 (see Figure 2). Truncated Charlier spectrum: $n = 40,\ \lambda_i = i$, $\omega_i = e^{-1}/(i-1)!$.

Example 3 (see Figure 3). Truncated Sturm–Liouville-like spectrum: $n = 100$, $\lambda_i = [\pi(i-0.5)]^2,\ \omega_i = 2$.

The curves drawn confirm the validity of the estimates.

We mention for comparison that the estimate of work [22] can be simplified for the case of frozen weights[6] and written as

$$\left\| \widetilde{J} - J \right\|_F \le L \left[\sum_{i=1}^{n} \left(\widetilde{\lambda}_i - \lambda_i \right)^2 \right]^{1/2},$$

where the index F denotes the Frobenius norm, and that the coefficient L for the matrices of Examples 1–3 equals approximately $7.3 \cdot 10^{93}$, $4.7 \cdot 10^{101}$, and $6.7 \cdot 10^{498}$, respectively.

Remark 2. These and other numerical experiments have shown that partial cancellation of relative errors may take part in (4.2) and similar formulae; in these cases the actual errors are less than those predicted by our theorems.

6. Influence of perturbation of weights. Since this is not a critical point, we shall keep the style (i.e., still use the Hankel determinants) rather than try to obtain the best possible estimates.

THEOREM 6.1. *If* $\max_{1 \le i \le n} \mathtt{RP}\, \omega_i \le \varepsilon$ *while* λ_i *are frozen, then*

$$\mathtt{RP}\, \beta_k \le \frac{1}{2} \overset{3}{\mathcal{D}} (4k\varepsilon), \tag{6.1}$$

[5] Fast computation was not an aim of ours, we just wanted to reliably obtain tables for drawing curves. There exist other computational methods, e.g., [7].

[6] The term $\|q - \tilde{q}\|_2$ vanishes in [22, formulae (3.20) and (3.23)].

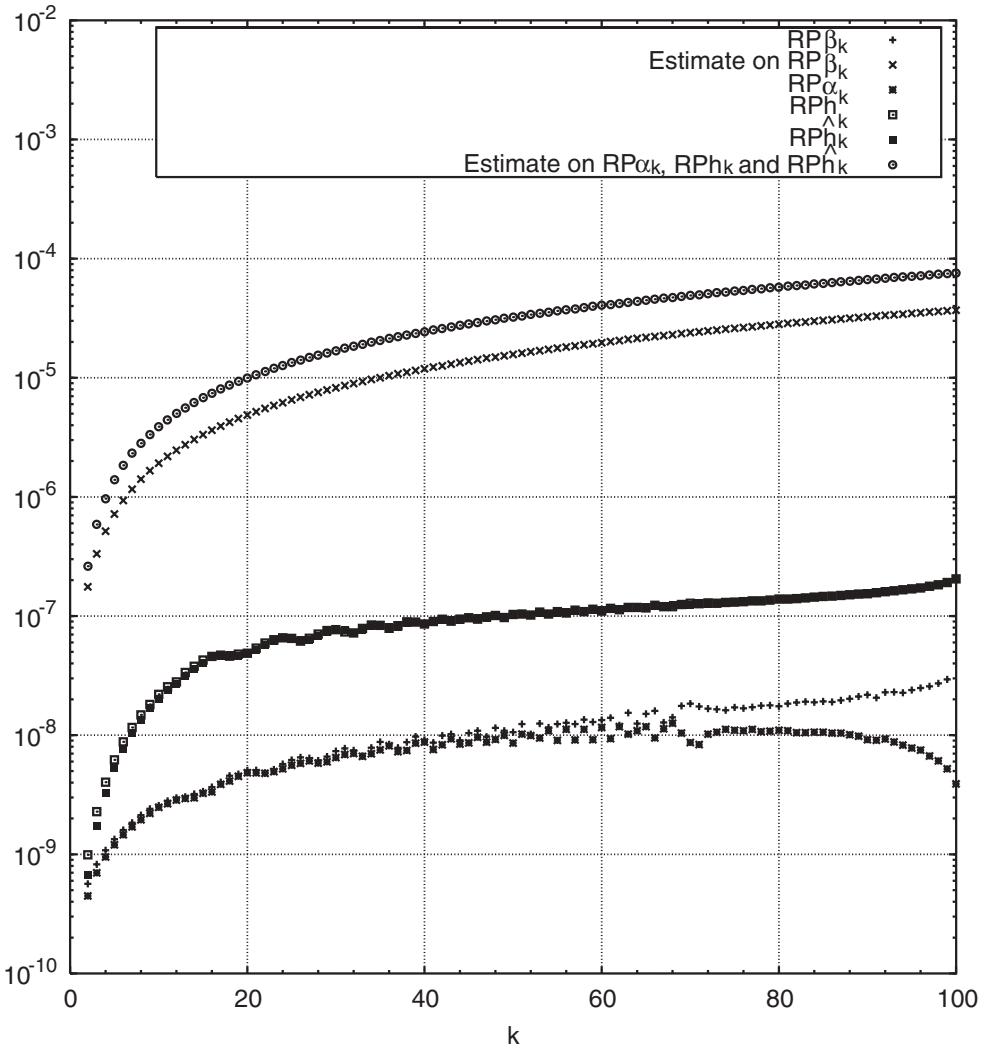


FIG. 1. *Equidistant spectrum with equal weights.*

and if $\lambda_1 > 0$, *also*

$$\mathtt{RP}\,\alpha_k \le \overset{3}{\mathcal{D}}(4k\varepsilon). \tag{6.2}$$

Proof. In the style of Lemmas 3.2 and 3.3 one can instantly show that $\mathtt{RP}\,H_k \le \mathcal{D}(k\varepsilon)$ and (when $\lambda_1 > 0$) $\mathtt{RP}\,G_k \le \mathcal{D}(k\varepsilon)$. In view of (4.2) and (4.4) this gives (6.1) and (6.2). □

7. A general perturbation scheme. The following assertion almost directly follows from formula (2.6).

PROPOSITION 7.1. *If* $\max_{1\le i\le n-1} \mathtt{AP}\,\mu_i \le \varepsilon$ *and* λ_j *remain unchanged, then*

$$\max_{1\le i\le n} \mathtt{RP}\,\omega_i \le \mathcal{D}\left[\frac{2\varepsilon}{d_2} + \frac{2\varepsilon(\log n + 1)}{d}\right]. \tag{7.1}$$

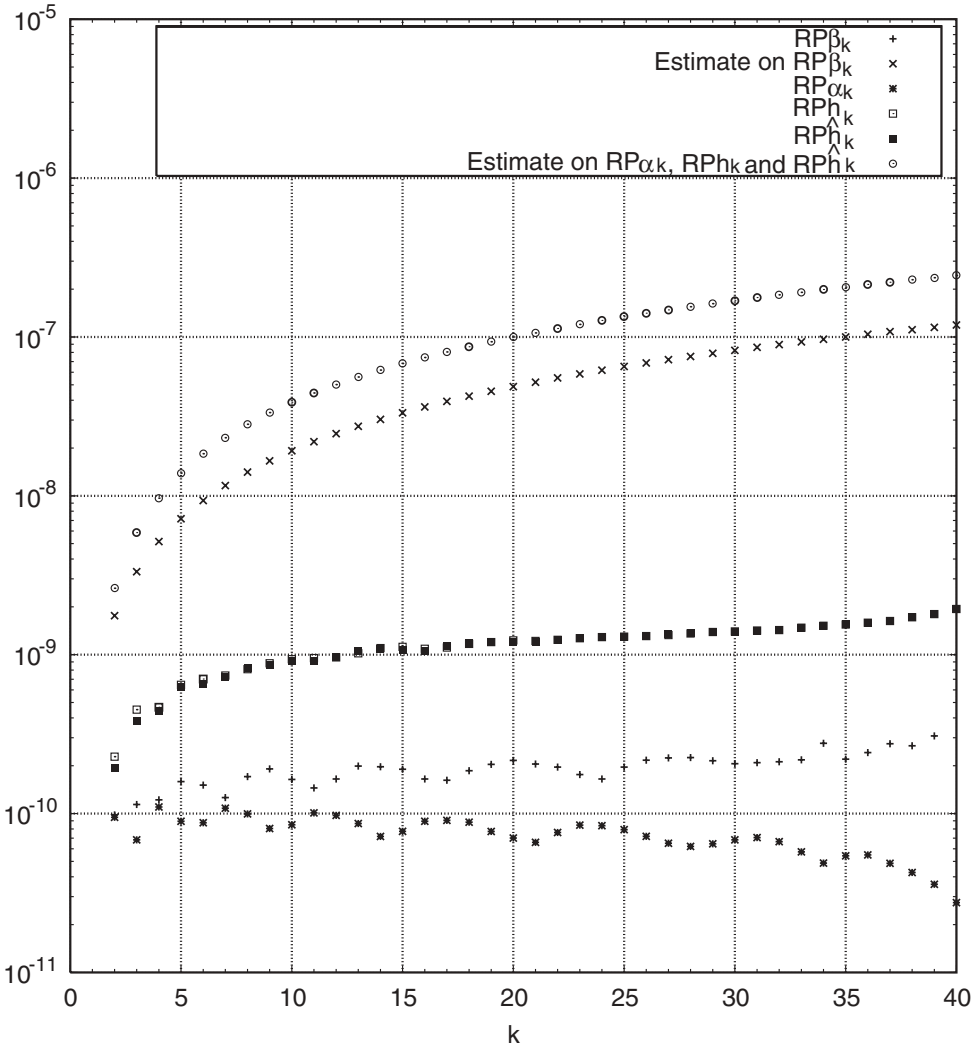


FIG. 2. *Truncated Charlier spectrum.*

If $\max_{1\le i\le n} \mathtt{AP}\,\lambda_i \le \varepsilon$ *and* μ_j *remain unchanged, then*

$$\max_{1\le i\le n} \mathtt{RP}\,\omega_i \le \overset{3}{\mathcal{D}} \left[\frac{2\varepsilon}{d_2} + \frac{6\varepsilon(\log n+1)}{d}\right]. \tag{7.2}$$

Proof. In the first case we note that

$$\sum_{j=1}^{n-1} \mathtt{RP}(\lambda_i-\mu_j) \le \frac{2\varepsilon}{d_2} + \frac{2\varepsilon}{d}\sum_{l=1}^{n-2}\frac{1}{l} \le \frac{2\varepsilon}{d_2} + \frac{2\varepsilon(\log n+1)}{d}.$$

In the second case again

$$\sum_{j=1}^{n-1} \mathtt{RP}(\lambda_i-\mu_j) \le \frac{2\varepsilon}{d_2} + \frac{2\varepsilon(\log n+1)}{d}$$

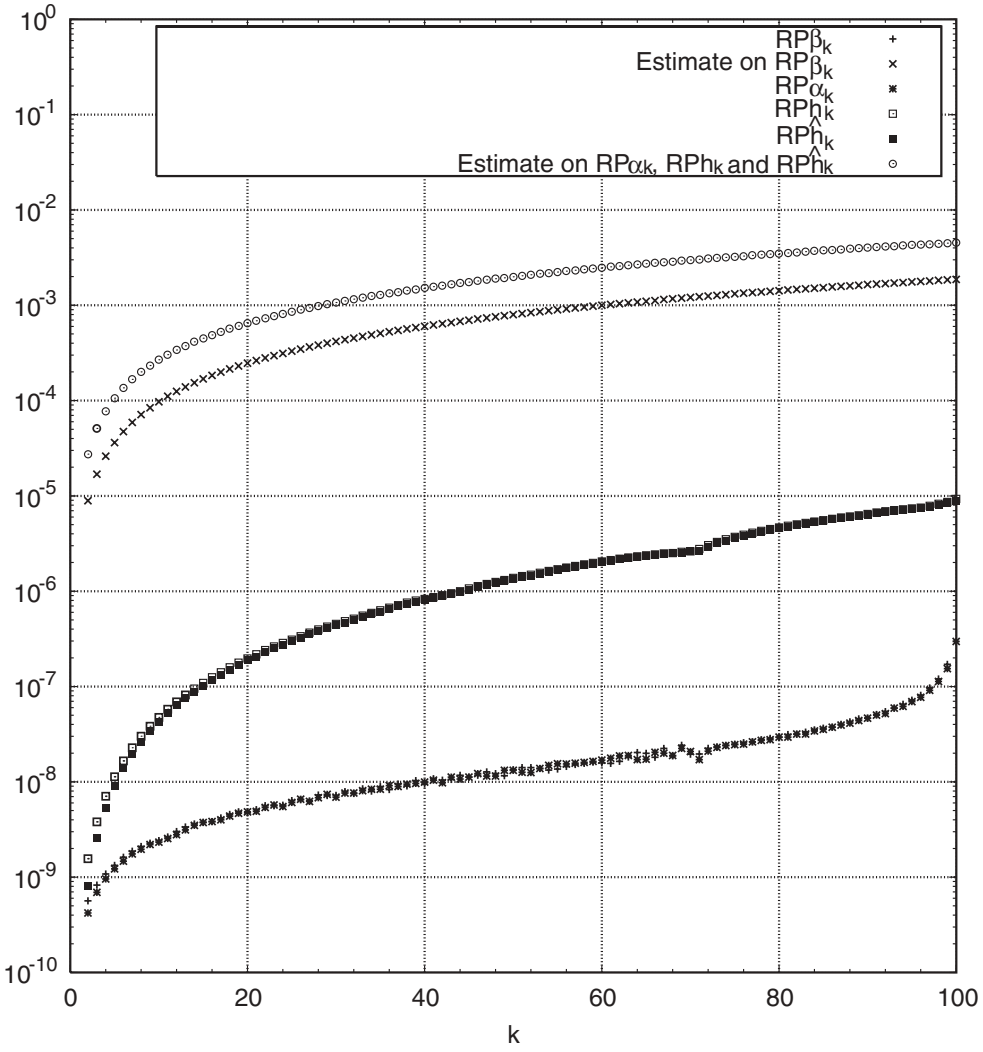


FIG. 3. *Truncated Sturm–Liouville-like spectrum.*

and

$$\sum_{1\le j\le n,\ j\ne i} \mathtt{RP}(\lambda_i-\lambda_j)\le \frac{4\varepsilon}{d}\sum_{l=1}^{n-2}\frac{1}{l}\le\frac{4\varepsilon(\log n+1)}{d}. \qquad \square$$

Any small simultaneous perturbation of λ_i and μ_j can be presented as a composition of special perturbations, which have already been studied. Denote by Λ, M, and Ω the sequences $(\lambda_1,\dots,\lambda_n)$, $(\mu_1,\dots,\mu_{n-1})$ and $(\omega_1,\dots,\omega_n)$, respectively. Imagine that we wish to move from $\Lambda^{(0)}$, $\mathrm{M}^{(0)}$, and the corresponding $\Omega^{(0)}$ to perturbed tuples $\widetilde{\Lambda}$, $\widetilde{\mathrm{M}}$, and $\widetilde{\Omega}$, also connected by (2.6). Suppose $\max_{1\le i\le n}\mathtt{AP}\,\lambda_i\le\varepsilon$ and $\max_{1\le j\le n-1}\mathtt{AP}\,\mu_j\le\varepsilon$.

1. Move from $\mathrm{M}=\mathrm{M}^{(0)}$ to $\mathrm{M}=\widetilde{\mathrm{M}}$, keeping $\Lambda=\Lambda^{(0)}$.[7] By means of (2.6)

[7]We may assume that the perturbed quantities $\widetilde{\mu}_j$ interlace not only the $\widetilde{\lambda}_i$, but also the λ_i: otherwise $\varepsilon\ge d_2$ and the estimates (7.3) are trivial.

calculate the weight tuple $\Omega = \Omega^{(1)}$, corresponding to $\Lambda = \Lambda^{(0)}$ and $\mathrm{M} = \widetilde{\mathrm{M}}$. Apply estimates (7.1) and (6.1)–(6.2) to analyze the effect of this perturbation on J:

$$\max_{1\le i\le n}\left|\frac{\omega_i^{(1)}}{\omega_i^{(0)}}-1\right| = O\left(\frac{\varepsilon\log n}{d_2}\right)$$

and

(7.3)
$$\frac{\beta_k\left(\Lambda^{(0)},\Omega^{(1)}\right)}{\beta_k\left(\Lambda^{(0)},\Omega^{(0)}\right)}-1 = O\left(\frac{\varepsilon k\log n}{d_2}\right), \qquad \frac{\alpha_k\left(\Lambda^{(0)},\Omega^{(1)}\right)}{\alpha_k\left(\Lambda^{(0)},\Omega^{(0)}\right)}-1 = O\left(\frac{\varepsilon k\log n}{d_2}\right),$$

the dependence on nodes and weights being explicitly indicated and the estimates for α_k being valid provided $\lambda_1 > 0$.

2. Compute $\Omega = \widetilde{\Omega}$ corresponding to $\Lambda = \widetilde{\Lambda}$ and $\mathrm{M} = \widetilde{\mathrm{M}}$. Bound the deviation of $\widetilde{\Omega}$ from $\Omega^{(1)}$ with the use of (7.2):

$$\max_{1\le i\le n}\left|\frac{\widetilde{\omega}_i}{\omega_i^{(1)}}-1\right| = O\left(\frac{\varepsilon\log n}{d_2}\right).$$

3. Jump from $\Omega = \Omega^{(1)}$ to $\Omega = \widetilde{\Omega}$, holding $\Lambda = \Lambda^{(0)}$. Exploit (6.1)–(6.2) to analyze the influence on J:

$$\frac{\beta_k(\Lambda^{(0)},\widetilde{\Omega})}{\beta_k\left(\Lambda^{(0)},\Omega^{(1)}\right)}-1 = O\left(\frac{\varepsilon k\log n}{d_2}\right), \qquad \frac{\alpha_k(\Lambda^{(0)},\widetilde{\Omega})}{\alpha_k\left(\Lambda^{(0)},\Omega^{(1)}\right)}-1 = O\left(\frac{\varepsilon k\log n}{d_2}\right). \tag{7.4}$$

4. Move from $\Lambda = \Lambda^{(0)}$ to $\Lambda = \widetilde{\Lambda}$, freezing $\Omega = \widetilde{\Omega}$. Apply estimates (4.1) and (4.3) to analyze the effect of this perturbation on J:

$$\frac{\beta_k(\widetilde{\Lambda},\widetilde{\Omega})}{\beta_k(\Lambda^{(0)},\widetilde{\Omega})}-1 = O\left(\frac{\varepsilon k\log k}{d}\right), \qquad \frac{\alpha_k(\widetilde{\Lambda},\widetilde{\Omega})}{\alpha_k(\Lambda^{(0)},\widetilde{\Omega})}-1 = O\left(\frac{\varepsilon k\log k}{d}+\frac{\varepsilon k}{\lambda_1}\right). \tag{7.5}$$

Combining the bounds (7.3)–(7.5), obtained at points 1, 3, and 4, we can estimate the total effect of the jump $(\Lambda^{(0)},\mathrm{M}^{(0)}) \longrightarrow (\widetilde{\Lambda},\widetilde{\mathrm{M}})$:

$$\mathtt{RP}\,\beta_k = O\left(\frac{\varepsilon k\log n}{d_2}\right), \qquad \mathtt{RP}\,\alpha_k = O\left(\frac{\varepsilon k\log n}{d_2}+\frac{\varepsilon k}{\lambda_1}\right).$$

We could write out concrete bounds with the use of $\overset{m}{\mathcal{D}}$, but we do not want to overcomplicate the formulae.

8. Unitary Hessenberg inverse eigenvalue problem. Any $n\times n$ upper Hessenberg unitary matrix with positive subdiagonal components is known to be uniquely presented as a product of Givens rotations/reflections

$$H(\gamma_1,\dots,\gamma_n) = \prod_{k=1}^{n-1}\begin{pmatrix} I_{k-1} & & & \\ & -\gamma_k & \sigma_k & \\ & \sigma_k & \bar\gamma_k & \\ & & & I_{n-k-1}\end{pmatrix}\cdot \mathrm{diag}(I_{n-1},-\gamma_n)$$

with

$$\gamma_k\in\mathbf{C}, \qquad |\gamma_k|<1 \quad (k=1,\dots,n-1), \qquad |\gamma_n|=1, \qquad \sigma_k=\sqrt{1-|\gamma_k|^2}. \tag{8.1}$$

The problem of constructing $H(\gamma_1, \dots, \gamma_n)$ from spectral data, including indication of applications, is discussed in [5, sect. 8].

The following uniqueness theorem for UHIEP was established in [2, 3].

THEOREM 8.1. *Given two sets $\{\lambda_1, \dots, \lambda_n\}$ and $\{\mu_1, \dots, \mu_n\}$ of strictly interlaced points on the unit circumference T, there exist a unique set of parameters γ_k obeying* (8.1) *and a unique α on T such that the spectrum of $H(\gamma_1, \dots, \gamma_n)$ is $\{\lambda_1, \dots, \lambda_n\}$ and the spectrum of $H(\alpha\gamma_1, \dots, \alpha\gamma_n)$ is $\{\mu_1, \dots, \mu_n\}$.*

We shall investigate the corresponding stability problem, using the same technique as for JIEP. The formula [2, Proposition 4.2]

$$\alpha = \prod_{i=1}^{n} \frac{\mu_i}{\lambda_i}$$

makes the analysis of perturbation of α trivial, so we shall concentrate on the parameters γ_k.

Note that $\alpha \neq 1$. Define a discrete positive measure τ on T with the nodes λ_i and the corresponding weights [2, the proof of Proposition 4.3]

$$\omega_i = \frac{1}{|1-\alpha|\,|\lambda_i|} \cdot \frac{\prod_{j=1}^{n} |\lambda_i - \mu_j|}{\prod_{1\le j\le n,\ j\neq i} |\lambda_i - \lambda_j|}. \tag{8.2}$$

The measure τ induces the system ϕ_k $(0 \le k \le n-1)$ of polynomials (Szegö polynomials) orthonormal on T and having positive leading coefficients (see [20, Chap. 11] and [17, Chap. 3]).

Introduce the moments $c_k = \int_T z^{-n}\, d\tau(z)$ and the determinants

$$D_k = \begin{vmatrix} c_0 & c_{-1} & \dots & c_{-k+1} \\ c_1 & c_0 & \dots & c_{-k+2} \\ \vdots & \vdots & \dots & \vdots \\ c_{k-1} & c_{k-2} & \cdots & c_0 \end{vmatrix} > 0, \quad E_k = \begin{vmatrix} c_{-1} & c_{-2} & \dots & c_{-k} \\ c_0 & c_{-1} & \dots & c_{-k+1} \\ \vdots & \vdots & \dots & \vdots \\ c_{k-2} & c_{k-3} & \cdots & c_{-1} \end{vmatrix} \in \mathbf{C}, \tag{8.3}$$
$$k = 0, \dots, n.$$

The determinant representation

$$\phi_k(z) = (D_k D_{k+1})^{-1/2} \begin{vmatrix} c_0 & c_{-1} & \dots & c_{-k} \\ c_1 & c_0 & \dots & c_{-k+1} \\ \vdots & \vdots & \dots & \vdots \\ c_{k-1} & c_{k-2} & \dots & c_{-1} \\ 1 & z & \dots & z^k \end{vmatrix}$$

follows

$$\phi_k(0) = \frac{(-1)^k E_k}{\sqrt{D_k D_{k+1}}} \tag{8.4}$$

and

$$\kappa_k \equiv \text{leading coefficient of } \phi_k = \sqrt{\frac{D_k}{D_{k+1}}}. \tag{8.5}$$

One has integral expressions

$$D_k = \frac{1}{k!}\int_T \cdots \int_T \prod_{1\le i<j\le k} |t_j - t_i|^2 \, d\tau(t_1)\dots d\tau(t_k), \tag{8.6}$$

$$E_k = \frac{1}{k!}\int_T \cdots \int_T \prod_{1\le i<j\le k} |t_j - t_i|^2 \prod_{i=1}^{k} t_i \cdot d\tau(t_1)\dots d\tau(t_k), \tag{8.7}$$

the former of which is provided by [20, sect. 11.1, reference to formula (2.2.11)], the latter following from the former with the use of analytical continuation by weights $(\omega_i \mapsto \lambda_i\omega_i)$.

LEMMA 8.2. *The Schur parameters γ_k possess the determinant representation*

$$\gamma_k = \frac{(-1)^k E_k}{D_k}, \qquad k = 1, \dots, n. \tag{8.8}$$

Proof. Formula [3, (2)] can be written as

$$\frac{\phi_k(z)}{\kappa_k} = z\frac{\phi_{k-1}(z)}{\kappa_{k-1}} + \gamma_k \frac{\phi^*_{k-1}(z)}{\kappa_{k-1}}, \qquad k = 1, \dots, n, \tag{8.9}$$

$$\frac{\phi_n(z)}{\kappa_n} \equiv \prod_{i=1}^{n}(z - \lambda_i),$$

where $\phi^*_{k-1}(z) = z^{k-1}\phi_{k-1}(1/z)$. Comparing (8.9) and formula [17, Chap. 3, sect. 1, (1.7)]

$$\frac{\phi_k(z)}{\kappa_k} = z\frac{\phi_{k-1}(z)}{\kappa_{k-1}} + \frac{\phi_k(0)}{\kappa_k}\cdot\frac{\phi^*_{k-1}(z)}{\kappa_{k-1}}, \qquad k = 1, \dots, n-1,$$

and taking into account (8.4) and (8.5), we obtain

$$\gamma_k = \frac{\phi_k(0)}{\kappa_k} = \frac{(-1)^k E_k}{D_k}. \tag{8.10}$$

In the case $k = n$ we have

$$\frac{(-1)^n E_n}{D_n} = (-1)^n \prod_{i=1}^{n} \lambda_i = \gamma_n \frac{\phi^*_{n-1}(0)}{\kappa_{n-1}} = \gamma_n. \qquad \square$$

Define the absolute separations

$$d = \min_{1\le i,j\le n,\ i\ne j} |\lambda_i - \lambda_j| > 0, \qquad d_2 = \min_{1\le i,j\le n} |\lambda_i - \mu_j| > 0. \tag{8.11}$$

THEOREM 8.3. *Let λ_i be perturbed such that $\mathtt{AP}\,\lambda_i \le \varepsilon$, and let the weights ω_i remain unchanged. Then*

$$\mathtt{AP}\,\gamma_k = O\left(\frac{k^2\varepsilon}{d}\right). \tag{8.12}$$

Proof. The use of (8.7) and (8.6) gives

$$\begin{aligned}\mathtt{AP}\, E_k &= \mathtt{AP}\,\frac{1}{k!}\sum_{(t_1,\dots,t_k),\ \{t_1,\dots,t_k\}\subseteq\{\lambda_1,\dots,\lambda_n\}}\ \prod_{1\le i<j\le k}|t_i-t_j|^2\prod_{i=1}^{k}[\tau(t_i)t_i]\\ &\le \frac{1}{k!}\sum_{(t_1,\dots,t_k),\ \{t_1,\dots,t_k\}\subseteq\{\lambda_1,\dots,\lambda_n\}}\ \mathtt{AP}\prod_{1\le i<j\le k}|t_i-t_j|^2\prod_{i=1}^{k}[\tau(t_i)t_i]\\ &\le O\left(\frac{k^2\varepsilon}{d}+k\varepsilon\right)\frac{1}{k!}\sum_{(t_1,\dots,t_k),\ \{t_1,\dots,t_k\}\subseteq\{\lambda_1,\dots,\lambda_n\}}\ \prod_{1\le i<j\le k}|t_i-t_j|^2\prod_{i=1}^{k}\tau(t_i)\\ &= O\left(\frac{k^2\varepsilon}{d}\right)D_k.\end{aligned}$$

Analogously,

$$\mathtt{RP}\, D_k = O\left(\frac{k^2\varepsilon}{d}\right).$$

Recalling (8.1) and (8.8), we finally derive

$$\begin{aligned}\mathtt{AP}\,\gamma_k &= \left|\frac{\widetilde{E}_k}{\widetilde{D}_k}-\frac{E_k}{D_k}\right| \le \frac{|\widetilde{E}_k-E_k|}{\widetilde{D}_k}+|E_k|\ |\widetilde{D}_k^{-1}-D_k^{-1}|\\ &\le \frac{|\widetilde{E}_k-E_k|}{D_k}\cdot\frac{D_k}{\widetilde{D}_k}+\left|1-\frac{D_k}{\widetilde{D}_k}\right| = O\left(\frac{k^2\varepsilon}{d}\right). \qquad \square\end{aligned}$$

Similar reasoning leads to the following theorem.

THEOREM 8.4. *Let ω_i be perturbed such that $\mathtt{RP}\,\omega_i \le \varepsilon$, and let us freeze the nodes λ_i. Then*

$$\mathtt{AP}\,\gamma_k = O(k\varepsilon). \tag{8.13}$$

Remark 3. The quantities γ_k are invariant with respect to a proportional change of weights ω_i, because the numerator E_k and denominator D_k in (8.8) are both homogeneous of degree k in a common multiple due to (8.3).

PROPOSITION 8.5. *If $\max_{1\le i\le n}\mathtt{AP}\,\lambda_i \le \varepsilon$ and μ_i are frozen, then*

$$\max_{1\le i\le n}\mathtt{RP}\,|1-\alpha|\omega_i = O\left(\frac{n\varepsilon}{d}+\frac{n\varepsilon}{d_2}\right). \tag{8.14}$$

If $\max_{1\le i\le n}\mathtt{AP}\,\mu_i \le \varepsilon$ and λ_i are frozen, then

$$\max_{1\le i\le n}\mathtt{RP}\,|1-\alpha|\omega_i = O\left(\frac{n\varepsilon}{d_2}\right). \tag{8.15}$$

Proof. Both assertions are straightforward consequences of (8.2) and (8.11). $\square$

An arbitrary small simultaneous perturbation of λ_i and μ_j can be analyzed analogously to what was done in section 7. Formulae (8.14), (8.15), and Remark 3, applied to the multiple $|1-\alpha|$, reduce estimating the result of a general eigenvalue perturbation to (8.12) and (8.13).

Note that we could estimate a relative perturbation of the so-called complimentary Schur parameters σ_k $(1 \le k \le n-1)$ (see (8.1)) owing to the representation

$$\sigma_k = \sqrt{1 - \left(\frac{|\phi_k(0)|}{\kappa_k}\right)^2} = \sqrt{1 - \frac{\kappa_k^2 - \kappa_{k-1}^2}{\kappa_k^2}} = \frac{\kappa_{k-1}}{\kappa_k} = \frac{\sqrt{D_{k-1}D_{k+1}}}{D_k}$$

(cf. (4.2); the formulae (8.10), [20, (11.3.6)], and (8.5) have been utilized).

Acknowledgments and conclusive remarks. I thank Liliana Borcea and Vladimir Druskin for focusing my attention on inverse spectral problems and for very useful discussions and organizational support; Liliana also corrected some points. In fact, this paper was originally motivated by their work [4] on the convergence analysis of the discrete inverse Sturm–Liouville problem. To complete the proof they need a sharp stability result conjectured with the help of numerical experiments, and it is stronger than the ones obtained here for JIEP (see Remark 2). This may be a subject of future research.

I thank W. Gautschi and G. Golub for having read a draft of the paper and their pieces of advice, B. Parlett for a useful talk and bibliographical support, and the three referees for numerous constructive remarks. I thank the Central Geophysical Expedition and Schlumberger–Doll Research for organizational support.

REFERENCES

[1] M. Abramowitz and J. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, National Bureau of Standards Applied Mathematics Series 55, US Government Printing Office, Washington, DC, 1964.

[2] G. Ammar, W. Gragg, and L. Reichel, *Constructing a unitary Hessenberg matrix from spectral data*, in Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms, G. H. Golub and P. Van Dooren, eds., NATO Adv. Sci. Inst. Ser. F Comp. Systems Sci. 70, Springer-Verlag, Berlin, 1991, pp. 385–395.

[3] G. S. Ammar and C. Y. He, *On an inverse eigenvalue problem for unitary Hessenberg matrices*, Linear Algebra Appl., 218 (1995), pp. 263–271.

[4] L. Borcea and V. Druskin, *Optimal finite difference grids for direct and inverse Sturm–Liouville problems*, Inverse Problems, 18 (2002), pp. 979–1001.

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

[6] V. Druskin and L. Knizhnerman, *Gaussian spectral rules for the three-point second differences:* I. *A two-point positive definite problem in a semi-infinite domain*, SIAM J. Numer. Anal., 37 (2000), pp. 403–422.

[7] H.-J. Fischer, *On generating orthogonal polynomials for discrete measures*, Z. Anal. Anwendungen, 17 (1998), pp. 183–205.

[8] F. Gantmakher, *The Theory of Matrices*, Vol. 1, Nauka, Moscow, 1988 (in Russian); English translation by K. A. Hirsch, AMS Chelsea Publishing, Providence, RI, 1998.

[9] W. Gautschi, *On generating orthogonal polynomials*, SIAM J. Sci. Statist. Comput., 3 (1982), pp. 289–317.

[10] W. Gautschi, *Computational aspects of orthogonal polynomials*, in Orthogonal Polynomials: Theory and Practice, Paul Nevai, ed., NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci., 294, Kluwer, Dordrecht, 1990, pp. 181–216.

[11] Ya. L. Geronimus, Addendum to the Russian ed. of [20], Fizmatgiz, Moscow, 1962.

[12] G. H. Golub and G. Meurant, *Matrices, moments and quadrature*, in Numerical Analysis 1993 (Dundee, 1993), Longman Sci. Tech., Harlow, 1994, pp. 105–156.

[13] O. H. Hald, *Inverse eigenvalue problems for Jacobi matrices*, Linear Algebra Appl., 14 (1976), pp. 63–85.

[14] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[15] H. Hochstadt, *On some inverse problems in matrix theory*, Arch. Math., 18 (1967), pp. 201–207.

[16] I. S. Kac and M. G. Krein, *On the spectral functions of the string*, Amer. Math. Soc. Transl., 103 (1974), pp. 19–102.

[17] E. M. Nikishin and V. N. Sorokin, *Rational Approximations and Orthogonality*, Nauka, Moscow, 1988 (in Russian); Transl. Math. Monogr. 92, AMS, Providence, RI, 1991, in English.

[18] B. N. Parlett, *Reduction to tridiagonal form and minimal realizations*, SIAM J. Matrix. Anal. Appl., 13 (1992), pp. 567–593.

[19] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Classics in Applied Mathematics, 20, SIAM, Philadelphia, 1998.

[20] G. Szegö, *Orthogonal Polynomials*, Amer. Math. Soc. Colloq. Publ. 23, AMS, Providence, RI, 1975.

[21] K. Vince Fernando and B. N. Parlett, *Accurate singular values and differential qd algorithms*, Numer. Math., 67 (1994), pp. 191–229.

[22] S.-F. Xu, *A stability analysis of the Jacobi matrix inverse eigenvalue problem*, BIT, 33 (1993), pp. 695–702.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 1, pp. 170–185

CONVEXITY AND ELASTICITY OF THE GROWTH RATE IN SIZE-CLASSIFIED POPULATION MODELS*

S. J. KIRKLAND†, M. NEUMANN‡, AND J. XU§

Abstract. This paper investigates both the convexity and elasticity of the growth rate of size-classsified population models. For an irreducible population projection matrix, we discuss the convexity properties of its Perron eigenvalue under perturbation of the vital rates, extending work of Kirkland and Neumann on Leslie matrices. We also provide nonnegative attainable lower bounds on the derivatives of the elasticity of the Perron eigenvalue under perturbation of the vital rates, sharpening, in the context of population projection matrices, the main result of Kirkland, Neumann, Ormes, and Xu.

Key words. size-classified models, group inverses, Perron eigenvalue, Perron eigenvector, elasticity, convexity

AMS subject classifications. 15A09, 15A18, 15A48, 92D25

DOI. 10.1137/S0895479802411031

1. Introduction.

In this paper we deal with a discrete time population model originally due to Lefkovitch [13]. The model is commonly known as the *size-classified model* for population growth. (In describing this model, we follow the notation and terminology used in Caswell [5, sect. 4.2].) We consider a population of organisms classified into n groups. In one time unit, for each $i = 2, \dots, n$, a member of the population in group i stays in group i with probability P_i, and for each $i = 1, \dots, n-1$, an individual in group i moves to group $i+1$ with probability G_i. Further, it is assumed that all members of the population are born into group 1, and that the fecundity rate for an individual in group i is F_i, $i = 1, \dots, n$.[1] Taken together, the quantities $F_1, \dots, F_n, G_1, \dots, G_{n-1}$ and $P_2, \dots, P_n$ are referred to as the *vital rates*.

Letting $x(t)$ be the n-vector whose ith entry is the number of individuals in group i at time t, $i = 1, \dots, n$, and assuming that the vital rates are independent of t, we arrive at the following fundamental relation:

$$x(t+1) = \begin{bmatrix} F_1 & F_2 & \cdots & \cdots & \cdots & F_n \\ G_1 & P_2 & 0 & \cdots & \cdots & 0 \\ 0 & G_2 & P_3 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & G_{n-1} & P_n \end{bmatrix} x(t) := Ax(t). \tag{1.1}$$

*Received by the editors July 16, 2002; accepted for publication (in revised form) by R. Nabben July 31, 2003; published electronically August 27, 2004.

http://www.siam.org/journals/simax/26-1/41103.html

†Department of Mathematics and Statistics, University of Regina, Regina, SK, S4S 0A2, Canada (kirkland@math.uregina.ca). The research of this author was supported in part by NSERC under grant OGP0138251.

‡Department of Mathematics, University of Connecticut, Storrs, CT 06269-3009 (neumann@math.uconn.edu). The work of this author was supported in part by NSF grant DMS0201333.

§Department of Mathematics and Statistics, University of West Florida, Pensacola, FL 32514 (jxu@uwf.edu).

[1] Strictly speaking, F_1 is not simply a fecundity rate but rather represents the contribution of individuals in group 1 at time t to the number of individuals in group 1 at time $t+1$. Thus F_1 combines the fecundity rate of those in group 1 with the probability of remaining in group 1 after one time unit.

The matrix A is known as the *projection matrix* for the population model. In the special case that $P_i = 0,\ i = 2, \ldots, n$, we have the much-studied *Leslie model*, where the ith group consists of the individuals whose ages are between $i-1$ and i time units, $i = 1, \ldots, n$, and where the maximum possible age is n time units. We remark that while the Leslie model has received much attention in modeling human populations (see [16] for example), the size-classsified model arises more naturally for other populations (see [5, p. 58], for example).

From (1.1), it follows that $x(t) = A^t x(0)$, so that the structure of the powers of A governs the growth of the population. In particular if A is *primitive*, that is, some power has all positive entries, then Perron–Frobenius theory applies.[2] Consequently, there is a positive number λ which is the eigenvalue of A having largest modulus, and corresponding right and left eigenvectors v and z^T, respectively, both of which have all positive entries; λ is known as the *Perron eigenvalue* for A while v and z^T are *right* and *left Perron eigenvectors*, respectively. It follows that if v is normalized so that $\mathbf{1}^T v = 1$ (where $\mathbf{1}$ denotes the all ones vector), then as $t \to \infty$, *x(t)* is asymptotic to a scalar multiple of $\lambda^T v$. Thus the eigenvalue λ can be interpreted as the (asymptotic) *growth rate* of the population, while the vector v can be thought of as the *stable population structure*. The latter term arises because $Av = \lambda v$, so that ratios of entries in v are unchanged under projection by A.

In this paper we shall be interested in the behavior of the growth rate λ as a single vital rate in A is perturbed. There are two natural ways by which we can measure the effect on the growth rate of a perturbation in the vital rates, namely *sensitivity analysis* and *elasticity analysis*.

Sensitivity analysis in population models consists mostly of considering the derivatives of the growth rate with respect to the vital rates (see [6], for example). Specifically, suppose that we have an $n \times n$ irreducible nonnegative matrix M with Perron eigenvalue λ_1. Let v and z^T be right and left Perron eigenvectors for M, respectively, normalized so that $z^T v = 1$. Then it follows from standard results (see Wilkinson [19] or Stewart [18], for example) that

$$\frac{\partial \lambda_1}{\partial m_{i,j}} \ = \ v_j z_i, \quad i,j = 1, \ldots, n. \tag{1.2}$$

In particular, since both v and z^T are positive vectors, $\frac{\partial \lambda_1}{\partial m_{i,j}} > 0$, for each i and j.

Formulae are also available for the second derivative of the Perron eigenvalue λ_1 with respect to $m_{i,j}$. In [8, 9], it is shown that if $Q^{\#}$ denotes the group (generalized) inverse[3] of the singular matrix $Q = \lambda_1 I - M$, then

$$\frac{\partial^2 \lambda_1}{\partial m_{i,j}^2} \ = \ 2 v_j z_i q_{j,i}^{\#}, \quad i,j = 1, \ldots, n. \tag{1.3}$$

From (1.3) we see that sign $(\frac{\partial^2 \lambda_1}{\partial m_{i,j}^2}) = \text{sign}(q_{j,i}^{\#})$, so that λ_1 is a convex or concave

[2] Applying a standard result from the theory of nonnegative matrices (see [3], for example), we find that the matrix A of (1.1) is primitive if and only if each of $G_1, \ldots, G_{n-1}$ and F_n is positive, and, in addition, either some P_i is positive or $\gcd\{j | F_j > 0\} = 1$.

[3] The group inverse of a matrix $B \in \mathbb{R}^{n,n}$, when it exists, is the unique matrix $X \in \mathbb{R}^{n,n}$ which satisfies the matrix equations $BXB = B$, $XBX = X$, and $BX = XB$. For background material on generalized inverses see Ben-Israel and Greville [2] and Campbell and Meyer [4]. Algorithms for computing X can be found in Anstriecher and Rothblum [1] and Hartwig [10]. We comment that in many of our works we have computed $Q^{\#}$ according to an explicit formula for it found by Meyer [14, p. 457]. Our numerical experience with this formula is very favorable.

function of $m_{i,j}$ according to whether $q^{\#}_{j,i}$ is positive or negative, respectively. Consequently, we see that $Q^{\#}$ carries qualitative information on the behavior of λ_1 as a function of $m_{i,j}$, while together v, z^T and $Q^{\#}$ can be used to quantify that behavior, and are key to the sensitivity analysis of λ_1. In section 3, we consider the class of projection matrices arising from (1.1) and provide an explicit formula for the corresponding matrix $Q^{\#}$. Analysis of the signs of certain entries of $Q^{\#}$ then yields qualitative information on the nature of the population growth rate as a function of the vital rates. These results generalize those of [11], which deals exclusively with Leslie matrices.

The elasticity analysis, on the other hand, concerns the proportional response of the growth rate to changes in the vital rates. Considering the projection matrix A of (1.1), we observe that the probabilities P_i and G_i are always bounded above by 1, while the fecundity rates F_j may well exceed 1. Because of such a difference in scale, some population biologists—for example, De Kroon et al. [7]—have suggested choosing elasticity analysis as an alternative to sensitivity analysis. Specifically, the elasticity of λ_1 with respect to the (i,j)th entry of M is defined as follows:

$$e_{i,j} := \frac{m_{i,j}}{\lambda_1}\frac{\partial \lambda_1}{\partial m_{i,j}}. \tag{1.4}$$

In [5, sect. 9.7], Caswell discusses the *convexity* of the growth rate by means of the sensitivity of the elasticities to changes in the vital rates. In particular, Caswell deduces from (1.4) that for $i,j,k,\ell = 1,\ldots,n$,

$$\frac{\partial e_{i,j}}{\partial m_{k,\ell}} = \frac{m_{i,j}}{\lambda_1}\frac{\partial^2 \lambda_1}{\partial m_{i,j}\partial m_{k,\ell}} - \frac{m_{i,j}}{\lambda_1^2}\frac{\partial \lambda_1}{\partial m_{k,\ell}}\frac{\partial \lambda_1}{\partial m_{i,j}} + \frac{\delta_{i,k}\delta_{j,\ell}}{\lambda_1}\frac{\partial \lambda_1}{\partial m_{i,j}}, \tag{1.5}$$

where $\delta_{p,q}$ is 1 or 0 according to whether $p = q$ or not. In section 4, we consider a projection matrix A arising from (1.1), and using the explicit formula for $Q^{\#}$, we provide a way of computing the derivatives $(\partial e_{i,j}/\partial a_{i,j})$ with respect to the vital rates. Lower bounds on these derivatives are also presented.

Throughout, we will assume basic knowledge of the theory of nonnegative matrices; some familiarity with matrix models for population growth is helpful, though not essential. For background on the former we refer the reader to [3, 17], while [5] provides an extensive discussion of the latter.

2. Preliminaries. Suppose that we have a population projection matrix A of the form arising in (1.1). If, in addition, each row of A sums to 1, we shall say that A is a *stochastic population projection matrix*. Thus, A is an $n \times n$ irreducible stochastic population projection matrix if and only if it can be written as

$$A = \left[\begin{array}{c|cccc} a_1 & a_2 & \cdots & \cdots & a_n \\ \hline 1-b_1 & b_1 & 0 & \cdots & 0 \\ 0 & 1-b_2 & b_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1-b_{n-1} & b_{n-1} \end{array}\right], \tag{2.1}$$

where $a_i \geq 0$ for $i = 1,\ldots,n-1$, $a_n > 0$, $\sum_{i=1}^{n} a_i = 1$, and $0 \leq b_i < 1$, for $i = 1,\ldots,n-1$.

In much of our subsequent development, it will be convenient to deal with matrices of the form (2.1), rather than with general population projection matrices. There is no loss of generality in this approach since the latter case can be reduced to the former, as we see from the following.

Remark 2.1. Suppose that $A \geq 0$ is an irreducible $n \times n$ matrix with Perron eigenvalue $\lambda > 0$ and a corresponding right Perron eigenvector $v = [v_1, \ldots, v_n]^T > 0$. Let $V = \mathrm{diag}(v_1, \ldots, v_n)$, and let $P = \frac{1}{\lambda} V^{-1} A V$. It is well known that P is irreducible and stochastic (see [3, p. 49], for example).

If u^T is the left Perron eigenvector for A normalized so that $u^T v = 1$, it is readily seen that $u^T V$ is the left Perron eigenvector for P, normalized so that $u^T V \mathbf{1} = 1$. Thus, if we set $S = v u^T$ (which by (1.2) carries information about the derivatives of λ with respect to the entries in A) and $\tilde{S} = \mathbf{1} u^T V = V^{-1} S V$ (which carries the corresponding information for P) we see that

$$\frac{\partial \lambda}{\partial a_{i,j}} = s_{j,i} = \frac{v_j}{v_i} \tilde{s}_{j,i}.$$

Further, it is easy to check that if $Q = \lambda I - A$ and $\tilde{Q} = I - P$, then

$$Q^{\#} = \frac{1}{\lambda} V \tilde{Q}^{\#} V^{-1}.$$

Putting these relations together with (1.3), we find that

$$\frac{\partial^2 \lambda}{\partial a_{i,j}^2} = \frac{2}{\lambda} \Big(\frac{v_j}{v_i} \Big)^2 \tilde{s}_{j,i} \tilde{q}_{j,i}^{\#}.$$

Thus we see that the first two derivatives of the Perron eigenvalue of A with respect to $a_{i,j}$ can be expressed in terms of the corresponding quantities for P. Note also that the signs of the second derivatives for A are the same as the corresponding signs of the second derivatives for P.

Using the information above in conjunction with (1.5), we find that

$$\frac{\partial e_{i,j}}{\partial a_{i,j}} = \frac{1}{\lambda} \Big(\frac{v_j}{v_i} \Big) \Big[2 p_{i,j} \tilde{s}_{j,i} \tilde{q}_{j,i}^{\#} - p_{i,j} \Big(\tilde{s}_{j,i} \Big)^2 + \tilde{s}_{j,i} \Big].$$

On the other hand, denoting the elasticities of P by $\tilde{e}_{i,j}$, it is clear from (1.2), (1.3), and (1.5) that

$$\frac{\partial \tilde{e}_{i,j}}{\partial p_{i,j}} \;=\; 2 p_{i,j} \tilde{s}_{j,i} \tilde{q}_{j,i}^{\#} - p_{i,j} \Big(\tilde{s}_{j,i} \Big)^2 + \tilde{s}_{j,i}$$

or, in matrix notation,

$$\Big[\frac{\partial e_{i,j}}{\partial a_{i,j}} \Big] \;=\; \frac{1}{\lambda} V^{-1} \Big[\frac{\partial \tilde{e}_{i,j}}{\partial p_{i,j}} \Big] V. \tag{2.2}$$

Thus the derivatives of the elasticities for A can be recovered from the corresponding derivatives of the elasticities for P via a diagonal similarity transformation using V. In particular, both derivative matrices have the same sign pattern.

Based on Remark 2.1, we shall consider in the sequel an $n \times n$ irreducible stochastic population projection matrix A of the form (2.1). Evidently, if A is such a matrix and $Q = I - A$, then we shall need to compute $Q^{\#}$ in order to apply formulae (1.3) and

(1.5). Fortunately, Meyer [15] provides a formula for $Q^{\#}$ for any irreducible stochastic matrix. Specifically, on letting w^T be the normalized left Perron eigenvector for A such that $w^T\mathbf{1} = 1$ and partitioning A as specified in (2.1), i.e.,

$$A = \begin{bmatrix} a_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}, \tag{2.3}$$

where $A_{2,2} \in \mathbb{R}^{n-1,n-1}$, then we have that

$$Q^{\#} = \left[\begin{array}{c|c} w_1 w_2^T Q_{2,2}^{-1}\mathbf{1} & -w_2^T Q_{2,2}^{-1}(I - \mathbf{1}w_2^T) \\ \hline -w_1(I - \mathbf{1}w_2^T)Q_{2,2}^{-1}\mathbf{1} & (I - \mathbf{1}w_2^T)Q_{2,2}^{-1}(I - \mathbf{1}w_2^T) \end{array}\right], \tag{2.4}$$

where we have partitioned w^T as $w^T = [w_1 \mid w_2^T]$, with $w_2 \in \mathbb{R}^{n-1}$, in conformity with the partitioning of A.

Set $b_0 := 0, s_0 := 0$, and $s_i := \sum_{j=1}^{i} a_j$, for $i = 1, \ldots, n-1$. It is straightforward to ascertain that the left Perron eigenvector for A is given by

$$w^T = \frac{1}{\displaystyle\sum_{i=0}^{n-1} \frac{1-s_i}{1-b_i}} \left[\frac{1-s_0}{1-b_0}, \ldots, \frac{1-s_{n-1}}{1-b_{n-1}}\right]. \tag{2.5}$$

We shall now proceed to compute the blocks of $Q^{\#}$ arising in (2.4). First, an easy calculation gives

$$Q_{2,2}^{-1} = \begin{bmatrix} \frac{1}{1-b_1} & 0 & \cdots & 0 \\ \frac{1}{1-b_1} & \frac{1}{1-b_2} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ \frac{1}{1-b_1} & \frac{1}{1-b_2} & \cdots & \frac{1}{1-b_{n-1}} \end{bmatrix}. \tag{2.6}$$

Thus

$$Q_{2,2}^{-1}\mathbf{1} = \left[\frac{1}{1-b_1}, \frac{1}{1-b_1} + \frac{1}{1-b_2}, \ldots, \frac{1}{1-b_1} + \cdots + \frac{1}{1-b_{n-1}}\right]^T. \tag{2.7}$$

Next, we find from (2.4) that

$$q_{1,1}^{\#} = w_1 w_2^T Q_{2,2}^{-1}\mathbf{1} = w_1^2 \sum_{i=1}^{n-1} \frac{1-s_i}{1-b_i} \sum_{m=1}^{i} \frac{1}{1-b_m}. \tag{2.8}$$

With $q_{1,1}^{\#}$ in hand, we see from (2.4) that the $(2,1)$-block of $Q^{\#}$ is given by

$$-w_1(I - \mathbf{1}w_2^T)Q_{2,2}^{-1}\mathbf{1} = -w_1 Q_{2,2}^{-1}\mathbf{1} + q_{1,1}^{\#}\mathbf{1}. \tag{2.9}$$

The remaining entries of $Q^{\#}$ can also be determined from (2.4) and (2.5), and therefore we obtain the following.

Observation 2.2. Let A be the irreducible stochastic population projection matrix given in (2.1) and let $Q = I - A$. For any $2 \le k \le n$ and for any $2 \le j \le n$,

$$q^{\#}_{j,k} = \frac{w_1}{1-b_{k-1}}\left[-(1-s_{k-1})\sum_{i=1}^{j-1}\frac{1}{1-b_i} - \sum_{i=k-1}^{n-1}\frac{1-s_i}{1-b_i} + w_1(1-s_{k-1})\sum_{i=1}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{i}\frac{1}{1-b_m}\right] \quad \text{if } j < k \tag{2.10}$$

and

$$q^{\#}_{j,k} = \frac{w_1}{1-b_{k-1}}\left[-(1-s_{k-1})\sum_{i=1}^{j-1}\frac{1}{1-b_i} - \sum_{i=k-1}^{n-1}\frac{1-s_i}{1-b_i} + w_1(1-s_{k-1})\sum_{i=1}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{i}\frac{1}{1-b_m}\right] + \frac{1}{1-b_{k-1}} \quad \text{if } j \ge k. \tag{2.11}$$

Furthermore, for $j = 1$ and for any $2 \le k \le n$,

$$q^{\#}_{1,k} = \frac{w_1}{1-b_{k-1}}\left[-\sum_{i=k-1}^{n-1}\frac{1-s_i}{1-b_i} + w_1(1-s_{k-1})\sum_{i=1}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{i}\frac{1}{1-b_m}\right]. \tag{2.12}$$

3. The structure of $Q^{\#}$ corresponding to population projection matrices. It follows from a result in [14, Thm. 4.1] that the diagonal entries of $Q^{\#}$ are positive; that fact is readily verified for the $(1,1)$ entry from (2.4). In fact, it turns out that the maximum entry in each column of $Q^{\#}$ is found on the diagonal (see [9, Thm. 3.1]). We continue in this vein by investigating certain monotonicity and sign properties found in the matrix $Q^{\#}$.

We begin with the following lemma.

LEMMA 3.1. *Let A be an irreducible stochastic population projection matrix given by* (2.1) *and let $Q = I - A$. Then for $j = 1, \ldots, n$, $q^{\#}_{j,1}$ is decreasing in j. Moreover, there exists an index $1 \le j_0 \le n-1$ such that $q^{\#}_{j_0,1} \ge 0 > q^{\#}_{j_0+1,1}$.*

Proof. From (2.7), (2.8), and (2.9) we see that for $j = 1, \ldots, n-1$,

$$\begin{aligned} q^{\#}_{j+1,1} &= q^{\#}_{1,1} - w_1 e_j^T Q_{2,2}^{-1}\mathbf{1} \\ &= w_1^2\left[\sum_{i=1}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{i}\frac{1}{1-b_m} - \sum_{i=0}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{j}\frac{1}{1-b_m}\right]. \end{aligned} \tag{3.1}$$

Thus, the entries of $Q^{\#}$ decrease as we proceed down its first column. From (2.8), we see that $q^{\#}_{1,1} > 0$, while from (3.1) it follows that

$$q^{\#}_{n,1} = -w_1^2\left[\sum_{i=1}^{n-2}\frac{1-s_i}{1-b_i}\sum_{m=i+1}^{n-1}\frac{1}{1-b_m} + \sum_{i=1}^{n-1}\frac{1}{1-b_i}\right] < 0.$$

Since $q^{\#}_{j,1}$ is decreasing in j, there must be an index $1 \leq j_0 \leq n-1$ such that $q^{\#}_{j_0,1} \geq 0 > q^{\#}_{j_0+1,1}$. ∎

Example 3.2. Lemma 3.1 shows that the entries in the first column of $Q^{\#}$ change sign at some index j_0 as we proceed down that column. This example illustrates the fact that such a sign change can take place at any position.

Fix an index j_0 with $1 \leq j_0 \leq n-1$, and construct an irreducible stochastic population projection matrix as in (2.1) by selecting parameters as follows: set $a_j = 0$, for all $1 \leq j \leq n-1$, $a_n = 1$, and $b_i = 0$, for all $i \neq j_0$. Finally, for a small positive number ϵ, let $b_{j_0} = 1 - \epsilon$. We claim that if ϵ is sufficiently small, then $q^{\#}_{j_0,1} > 0 > q^{\#}_{j_0+1,1}$.

From (3.1) we see that for $0 < \epsilon < 1$, we have that

$$\begin{aligned} q^{\#}_{j_0,1} &= w_1^2 \left[\sum_{i=1}^{j_0-1} i + \frac{1}{\epsilon}\left(j_0 - 1 + \frac{1}{\epsilon}\right) \right. \\ &\qquad \left. + \sum_{i=j_0+1}^{n-1} \left(i - 1 + \frac{1}{\epsilon}\right) - (j_0 - 1)\left(n - 1 + \frac{1}{\epsilon}\right) \right] \\ &= w_1^2 \left[\sum_{i=1}^{j_0-1} i + \frac{1}{\epsilon^2} + \sum_{i=j_0+1}^{n-1} \left(i - 1 + \frac{1}{\epsilon}\right) - (j_0 - 1)(n-1) \right], \end{aligned}$$

which is positive for all sufficiently small $\epsilon > 0$.

Next, consider $q^{\#}_{j_0+1,1}$. Again, by (3.1), we have that

$$\begin{aligned} q^{\#}_{j_0+1,1} &= w_1^2 \left[\sum_{i=1}^{j_0-1} i + \frac{1}{\epsilon}\left(j_0 - 1 + \frac{1}{\epsilon}\right) \right. \\ &\qquad \left. + \sum_{i=j_0+1}^{n-1} \left(i - 1 + \frac{1}{\epsilon}\right) - \left(j_0 - 1 + \frac{1}{\epsilon}\right)\left(n - 1 + \frac{1}{\epsilon}\right) \right] \\ &= w_1^2 \left[\sum_{i=1}^{j_0-1} i + \sum_{i=j_0+1}^{n-1} (i-1) - \frac{j_0}{\epsilon} - (j_0 - 1)(n-1) \right], \end{aligned}$$

which is negative for all sufficiently small $\epsilon > 0$.

Paralleling Lemma 3.1, our next result establishes a monotonicity result for entries in the other columns of $Q^{\#}$.

THEOREM 3.3. *Let A be an irreducible stochastic population projection matrix as given in* (2.1) *and let $Q = I - A$. Fix any $2 \leq k \leq n$. Then, $q^{\#}_{j-1,k} > q^{\#}_{j,k}$ for $j = 2, \ldots, k-1$ and for $j = k+1, \ldots, n$.*

Proof. Comparing (2.12) with (2.10) shows that $q^{\#}_{1,k} > q^{\#}_{2,k}$. For $j = 3, \ldots, k-1$, the fact that $q^{\#}_{j-1,k} > q^{\#}_{j,k}$ now follows from (2.10), while for $j = k+1, \ldots, n$, the corresponding inequality follows from (2.11). ∎

We next consider the behavior of the entries of $Q^{\#}$ along the superdiagonal.

THEOREM 3.4. *Let A be an irreducible stochastic population projection matrix as in* (2.1) *and let $Q = I - A$. If $q^{\#}_{k,k+1} \geq 0$ for some $k = 2, \ldots, n-1$, then $q^{\#}_{j,j+1} \geq 0$, for $1 \leq j \leq k-1$.*

Proof. We note that from (2.10) and (2.12), it follows readily that for $k = 1, \ldots, n-1$,

$$q^{\#}_{k,k+1} = \frac{w_1}{1-b_k}\left[-(1-s_k)\sum_{i=1}^{k-1}\frac{1}{1-b_i} + w_1(1-s_k)\sum_{i=1}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{i}\frac{1}{1-b_m} - \sum_{i=k}^{n-1}\frac{1-s_i}{1-b_i}\right]. \tag{3.2}$$

From (3.2) we note that $q^{\#}_{j,j+1} \geq 0$ if and only if

$$w_1\sum_{i=1}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{i}\frac{1}{1-b_m} \geq \sum_{i=1}^{j-1}\frac{1}{1-b_i} + \frac{1}{1-s_j}\sum_{i=j}^{n-1}\frac{1-s_i}{1-b_i}. \tag{3.3}$$

Since $q^{\#}_{k,k+1} \geq 0$, we see that

$$\begin{aligned} w_1\sum_{i=1}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{i}\frac{1}{1-b_m} &\geq \sum_{i=1}^{k-1}\frac{1}{1-b_i} + \frac{1}{1-s_k}\sum_{i=k}^{n-1}\frac{1-s_i}{1-b_i} \\ &= \sum_{i=1}^{k-2}\frac{1}{1-b_i} + \frac{1}{1-b_{k-1}} + \frac{1}{1-s_k}\sum_{i=k}^{n-1}\frac{1-s_i}{1-b_i} \\ &\geq \sum_{i=1}^{k-2}\frac{1}{1-b_i} + \frac{1}{1-s_{k-1}}\sum_{i=k-1}^{n-1}\frac{1-s_i}{1-b_i} \end{aligned}$$

since $s_k \geq s_{k-1}$. Our claim now follows by using an inductive argument. ∎

Remark 3.5. Note that from (3.2), $q^{\#}_{n-1,n} < 0$ if and only if

$$\sum_{i=1}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{i}\frac{1}{1-b_m} < \sum_{i=0}^{n-1}\frac{1-s_i}{1-b_i}\sum_{i=1}^{n-1}\frac{1}{1-b_i}.$$

This inequality is readily seen to be equivalent to the following:

$$-\sum_{i=1}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=i+1}^{n-1}\frac{1}{1-b_m} < \sum_{i=1}^{n-1}\frac{1}{1-b_i}. \tag{3.4}$$

As (3.4) always holds, we conclude that $q^{\#}_{n-1,n} < 0$.

Example 3.6. From Theorem 3.4 and Remark 3.5, we see that either the entire superdiagonal of $Q^{\#}$ is negative or, as we proceed down the superdiagonal, there is an index $2 \leq k \leq n-1$, where a sign change takes place, i.e., where $q^{\#}_{k-1,k} \geq 0 > q^{\#}_{k,k+1}$. In this example, we show that it is possible for the sign change to take place at any such index k and that it is possible for the entire superdiagonal to be negative.

Fix an index k with $2 \leq k \leq n-1$ and construct an irreducible stochastic population projection matrix as in (2.1) by selecting parameters as follows: for a small positive number ϵ, set $a_1 = \cdots = a_{k-1} = 0$, $a_k = 1-\epsilon$, $a_{k+1} = \cdots = a_{n-1} = 0$, and $a_n = \epsilon$, and let $b_1 = \cdots = b_k = 0$, and $b_{k+1} = \cdots = b_{n-1} = 1-\epsilon$. Note that

$1 - s_i = 1$ if $i < k$, while $1 - s_i = \epsilon$ if $i \geq k$. Also $(1 - s_i)/(1 - b_i) = 1$ if $i \neq k$, while $(1 - s_k)/(1 - b_k) = \epsilon$. It now follows that $w_1 = 1/(n - 1 + \epsilon)$. Applying (3.2), we find that

$$q^{\#}_{k-1,k} = w_1 \left\{ -\frac{(k-1)(k-2)}{2} + \frac{1}{n-1+\epsilon}\left[\frac{k(k-1)}{2} + \epsilon k + \frac{n-k-1}{\epsilon}\right] - (n-k) - \epsilon \right\},$$

which is readily seen to be positive for all sufficiently small $\epsilon > 0$. We also find from (3.2) that

$$q^{\#}_{k,k+1} = w_1 \left\{ -\epsilon\frac{k(k-1)}{2} + \frac{\epsilon}{n-1+\epsilon}\left[\frac{k(k-1)}{2} + \epsilon k + \frac{n-k-1}{\epsilon}\right] - (n-k) - \epsilon \right\},$$

and it is straightforward to show that this expression is negative for all sufficiently small positive ϵ. Finally, we note that the analogous construction for $k = 1$ and small $\epsilon > 0$ yields a matrix such that $q^{\#}_{1,2} < 0$, so that the entire superdiagonal of $Q^{\#}$ is negative.

Example 3.7. While $Q^{\#}$ does exhibit some monotonic behavior as we proceed down a column, there is, in general, no such monotonic behavior as we proceed down the superdiagonal or down the diagonal, as the following example illustrates. Let

$$A = \begin{bmatrix} 0.1139 & 0.2626 & 0.2574 & 0.1152 & 0.2509 \\ 0.9421 & 0.0579 & 0 & 0 & 0 \\ 0 & 0.6471 & 0.3529 & 0 & 0 \\ 0 & 0 & 0.1868 & 0.8132 & 0 \\ 0 & 0 & 0 & 0.9901 & 0.0099 \end{bmatrix}.$$

Then on using Matlab to compute $Q^{\#}$, we find that

$$Q^{\#} = \begin{bmatrix} 0.8165 & -0.0860 & -0.1726 & -0.7148 & 0.1569 \\ 0.6091 & 0.7803 & -0.3724 & -1.1213 & 0.1043 \\ 0.3071 & 0.4963 & 0.8819 & -1.7130 & 0.0278 \\ -0.7390 & -0.4876 & -0.1259 & 1.5898 & -0.2373 \\ -0.9364 & -0.6733 & -0.3161 & 1.2031 & 0.7227 \end{bmatrix}.$$

Thus, while there is structure to the signs of the entries on the diagonal and superdiagonal of $Q^{\#}$, those entries may not follow a monotonic pattern.

4. Elasticity of the Perron eigenvalue for a population projection matrix. Suppose that we have an irreducible stochastic population projection matrix A, as given in (2.1). In this section we develop formulae for the derivatives of the elasticity of its Perron eigenvalue as functions of the vital rates, and present lower bounds on those derivatives. We note that in [12] it is shown that for any $n \times n$ irreducible nonnegative matrix M, the elasticity of its Perron eigenvalue with respect to $m_{i,j}$ is a differentiable nondecreasing function of $m_{i,j}$. Thus, on letting $e_{i,j}$ be the elasticity of the Perron eigenvalue for A with respect to $a_{i,j}$, we have that

$$\frac{\partial e_{i,j}}{\partial a_{i,j}} \geq 0 \quad \forall\, i,j = 1,\ldots,n. \tag{4.1}$$

We then see that for matrices arising in the size-classified model, our lower bounds in this section generalize (4.1).

Note that for any pair (i,j), if we substitute (1.3) into (1.5), we find that

$$\frac{\partial e_{i,j}}{\partial a_{i,j}} = w_i\left(2a_{i,j}q^{\#}_{j,i} - a_{i,j}w_i + 1\right). \tag{4.2}$$

We begin by discussing $\partial e_{i,i}/\partial a_{i,i},\ i = 1,\dots,n$. Since $q^{\#}_{i,i} > 0$, for each such i, we find that

$$\frac{\partial e_{1,1}}{\partial a_{1,1}} \geq 1 - a_1 w_1 \tag{4.3}$$

and that

$$\frac{\partial e_{i,i}}{\partial a_{i,i}} \geq 1 - b_{i-1}w_i, \qquad i = 2,\dots,n. \tag{4.4}$$

Evidently, each $\partial e_{i,i}/\partial a_{i,i}$ is positive, as anticipated by (4.1).

Remark 4.1. We note that $\partial e_{i,i}/\partial a_{i,i}$ can be arbitrarily close to 0. To see this consider the matrix of (2.1) corresponding to the following selection of parameters: suppose that $\epsilon \in (0,1)$, and let $a_i = 1-\epsilon,\ a_n = \epsilon,\ b_{i-1} = 1-\epsilon$, and $a_j = 0,\ j \neq i,\ b_j = 0,\ j \neq i-1$. We find that $w_i = 1/[1+(j-1)\epsilon + (n-j)\epsilon^2]$ and $w_1 = \epsilon/[1+(j-1)\epsilon+(n-j)\epsilon^2]$. Also, applying (2.11), it follows that

$$\begin{aligned} q^{\#}_{i,i} = \frac{1}{1+(j-1)\epsilon+(n-j)\epsilon^2}&\left\{-(j-2) - \frac{2}{\epsilon} - \epsilon(n-j)\right. \\ &+\frac{\epsilon}{1+(j-1)\epsilon+(n-j)\epsilon^2}\left[\frac{(j-1)(j-2)}{2} + \frac{1}{\epsilon}\left(j-2+\frac{1}{\epsilon}\right) + (n-j)\epsilon\right]\Bigg\} \\ &+\frac{1}{\epsilon}. \end{aligned}$$

Evidently, when $\epsilon \to 0^+$ we see that $q^{\#}_{i,i} \to 0$, while $a_{i,i}w_i \to 1$. Thus, we observe that by choosing a sufficiently small $\epsilon > 0$, $\partial e_{i,i}/\partial a_{i,i}$ can be made arbitrarily close to 0.

Next we consider the derivatives of elasticities corresponding to the first row of A. Fix an index $1 \leq j \leq n-1$ and consider $\partial e_{1,j+1}/\partial a_{1,j+1}$. From (3.1) we see that

$$\begin{aligned} \frac{1}{w_1}\frac{\partial e_{1,j+1}}{\partial a_{1,j+1}} &= 2a_{j+1}w_1^2\left[\sum_{i=1}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{i}\frac{1}{1-b_m} - \sum_{i=0}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{j}\frac{1}{1-b_m}\right] \\ &\quad -a_{j+1}w_1 + 1, \end{aligned}$$

so that

$$\begin{aligned} \frac{1}{w_1^3}\frac{\partial e_{1,j+1}}{\partial a_{1,j+1}} &= 2a_{j+1}\left[\sum_{i=1}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{i}\frac{1}{1-b_m} - \sum_{i=0}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{j}\frac{1}{1-b_m}\right] \\ &\quad - a_{j+1}\sum_{i=0}^{n-1}\frac{1-s_i}{1-b_i} + \left[\sum_{i=0}^{n-1}\frac{1-s_i}{1-b_i}\right]^2. \end{aligned} \tag{4.5}$$

We define f_{j+1} by

$$f_{j+1}(a_1,\dots,a_n,b_1,\dots,b_{n-1}) := \frac{1}{w_1^3}\frac{\partial e_{1,j+1}}{\partial a_{1,j+1}}. \tag{4.6}$$

To derive the lower bound on f_{j+1}, we need the following two technical lemmas whose proofs are available from the authors.

LEMMA 4.2. *Let A be an irreducible stochastic population projection matrix as given in* (2.1) *and let $Q = I - A$. Then $\partial f_{j+1}/\partial b_k > 0$, for each $k = 1,\dots,n-1$.*

LEMMA 4.3. *Suppose that $0 \le s_1 \le s_2 \le \cdots \le s_{k+1} \le 1$. Then*

$$\left[\sum_{j=1}^{k}(1-s_j)\right]^2 \ge (1-s_{k+1})\sum_{j=1}^{k}(1-s_j)(2k+1-2j), \tag{4.7}$$

and the equality holds if and only if $s_1 = s_2 = \cdots = s_{k+1}$.

Lemmas 4.2 and 4.3 lead us to the result below.

THEOREM 4.4. *Let A be an irreducible stochastic population projection matrix as given in* (2.1)*, let $Q = I - A$, and let f_{j+1} be given by* (4.6)*. Then*

$$f_{j+1} \ge (1-s_{j+1})\sum_{i=0}^{j-1}(1-s_i)(2j-2i+1). \tag{4.8}$$

In particular, $f_{j+1} \ge 0$, with equality holding if and only if $a_i = b_i = 0,\ 1 \le i \le n-1$, and $j+1 = n$.

Proof. By Lemma 4.2, it suffices to show that the inequality holds for $f_{j+1}(a_1,\dots,a_n,0,\dots,0)$. But in that case we have that

$$\begin{aligned}
f_{j+1} &= 2a_{j+1}\left[\sum_{i=1}^{n-1}(1-s_i)i - \sum_{i=0}^{n-1}(1-s_i)j\right] - a_{j+1}\sum_{i=0}^{n-1}(1-s_i) \\
&\quad + \left[\sum_{i=0}^{n-1}(1-s_i)\right]^2 \\
&= -a_{j+1}\sum_{i=0}^{j-1}(1-s_i)(2j-2i+1) - a_{j+1}\sum_{i=j+1}^{n-1}(1-s_i)(2j-2i+1) \\
&\quad - a_{j+1}(1-s_j) + \left[\sum_{i=0}^{n-1}(1-s_i)\right]^2 \\
&\ge -a_{j+1}\sum_{i=0}^{j-1}(1-s_i)(2j-2i+1) - a_{j+1}(1-s_j) + \left[\sum_{i=0}^{j}(1-s_i)\right]^2 \\
&= -a_{j+1}\sum_{i=0}^{j-1}(1-s_i)(2j-2i+1) - a_{j+1}(1-s_j) + \left[\sum_{i=0}^{j-1}(1-s_i)\right]^2 \\
&\quad +2\sum_{i=0}^{j-1}(1-s_i) + 1,
\end{aligned}$$

where the inequality follows from the fact that $1 - s_i - a_{j+1} \geq 0$, for each $1 \leq i \leq j$.

But then, using Lemma 4.3, we obtain that

$$\begin{aligned} f_{j+1} &\geq -a_{j+1}\sum_{i=0}^{j-1}(1-s_i)(2j-2i+1) + (1-s_j)\sum_{i=0}^{j-1}(1-s_i)(2j-2i+1) \\ &\quad +2\sum_{i=0}^{j-1}(1-s_i) + 1 - a_{j+1}(1-s_j) \\ &\geq (1-s_j-a_{j+1})\sum_{i=0}^{j-1}(1-s_i)(2j-2i+1). \end{aligned} \tag{4.9}$$

Note that if $f_{j+1} = 0$, then necessarily $a_{j+1} = 1$, which implies that $j+1 = n$ and $a_i = 0,\ 1 \leq i \leq n-1$.

In addition, if $a_1 = \cdots = a_{n-1} = 0$ and $a_n = 1$, we readily see that $f_{j+1} = 0$. □

Finally, we consider the derivatives of elasticities corresponding to subdiagonal entries of A. Applying (3.2) and (4.2), we have that

$$\begin{aligned} \frac{1}{w_{k+1}}\frac{\partial e_{k+1,k}}{\partial a_{k+1,k}} &= 2a_{k+1,k}q^{\#}_{k,k+1} - a_{k+1,k}w_{k+1} + 1 \\ &= 2(1-b_k)q^{\#}_{k,k+1} - (1-b_k)w_1\frac{1-s_k}{1-b_k} + 1 \\ &= 2w_1\left[-(1-s_k)\sum_{i=1}^{k-1}\frac{1}{1-b_i}\right. \\ &\quad \left. +w_1(1-s_k)\sum_{i=1}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{i}\frac{1}{1-b_m} - \sum_{i=k}^{n-1}\frac{1-s_i}{1-b_i}\right] - (1-s_k)w_1 + 1 \\ &= w_1^2\left\{-2(1-s_k)\sum_{i=1}^{k-1}\frac{1}{1-b_i}\sum_{i=0}^{n-1}\frac{1-s_i}{1-b_i}\right. \\ &\quad +2(1-s_k)\sum_{i=1}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{i}\frac{1}{1-b_m} - 2\sum_{i=k}^{n-1}\frac{1-s_i}{1-b_i}\sum_{i=0}^{n-1}\frac{1-s_i}{1-b_i} \\ &\quad \left. -(1-s_k)\sum_{i=0}^{n-1}\frac{1-s_i}{1-b_i} + \left[\sum_{i=0}^{n-1}\frac{1-s_i}{1-b_i}\right]^2\right\} \\ &= w_1^2\left\{2(1-s_k)\sum_{i=1}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{i}\frac{1}{1-b_m} - 2(1-s_k)\sum_{i=1}^{k-1}\frac{1}{1-b_i}\sum_{i=0}^{n-1}\frac{1-s_i}{1-b_i}\right. \\ &\quad \left. -(1-s_k)\sum_{i=0}^{n-1}\frac{1-s_i}{1-b_i} + \left[\sum_{i=0}^{k-1}\frac{1-s_i}{1-b_i}\right]^2 - \left[\sum_{i=k}^{n-1}\frac{1-s_i}{1-b_i}\right]^2\right\}. \end{aligned}$$

Now we define g_k by

$$\begin{aligned} & g_k(a_1,\dots,a_{n-1},b_1,\dots,b_{n-1}) \\ & := 2(1-s_k)\sum_{i=1}^{n-1}\frac{1-s_i}{1-b_i}\sum_{m=1}^{i}\frac{1}{1-b_m} - 2(1-s_k)\sum_{i=1}^{k-1}\frac{1}{1-b_i}\sum_{i=0}^{n-1}\frac{1-s_i}{1-b_i} \\ & -(1-s_k)\sum_{i=0}^{n-1}\frac{1-s_i}{1-b_i} + \left[\sum_{i=0}^{k-1}\frac{1-s_i}{1-b_i}\right]^2 - \left[\sum_{i=k}^{n-1}\frac{1-s_i}{1-b_i}\right]^2. \end{aligned} \tag{4.10}$$

To establish the lower bound on g_k, we need the next two technical lemmas whose proofs are also available from the authors.

LEMMA 4.5. *Let A be an irreducible stochastic population projection matrix as given in* (2.1)*, let $Q = I - A$ and let g_k be given by* (4.10)*. Then $\partial g_k/\partial b_j > 0$, for $b_j \in [0,1)$, $1 \le j \le n-1$.*

LEMMA 4.6. *Suppose that $0 \le s_1 \le s_2 \le \cdots \le s_k \le 1$. Then*

$$(1-s_1)\sum_{j=1}^{k}(1-s_j)(2j-1) \;\ge\; \left[\sum_{j=1}^{k}(1-s_j)\right]^2, \tag{4.11}$$

and the equality holds if and only if $s_1 = s_2 = \cdots = s_k$.

Lemmas 4.5 and 4.6 lead us to the following result.

THEOREM 4.7. *Let A be an irreducible stochastic population projection matrix as given in* (2.1)*, let $Q = I - A$ and let g_k be given by* (4.10)*. Then*

$$\begin{aligned} g_k \ge{}& (1-s_k)\sum_{i=k}^{n-1}(1-s_i)(2i-2k+1) - \left[\sum_{i=k}^{n-1}(1-s_i)\right]^2 \\ & + \left[\sum_{i=0}^{k-1}(1-s_i)\right]^2 - (1-s_k)\sum_{i=0}^{k-1}(1-s_i)(2k-2i-1). \end{aligned} \tag{4.12}$$

In particular, $g_k \ge 0$, with equality if and only if $a_1 = \cdots = a_{n-1} = 0$, $a_n = 1$, and $b_1 = \cdots = b_{n-1} = 0$.

Proof. By Lemma 4.5, we see that

$$g_k(a_1,\dots,a_{n-1},b_1,\dots,b_{n-1}) \ge g_k(a_1,\dots,a_{n-1},0,\dots,0).$$

Note that when $b_1 = \cdots = b_{n-1} = 0$, we have that

$$\begin{aligned} g_k \;=\;& 2(1-s_k)\sum_{i=1}^{n-1}(1-s_i)i - 2(1-s_k)(k-1)\sum_{i=0}^{n-1}(1-s_i) \\ & -(1-s_k)\sum_{i=0}^{n-1}(1-s_i) + \left[\sum_{i=0}^{k-1}(1-s_i)\right]^2 - \left[\sum_{i=k}^{n-1}(1-s_i)\right]^2 \\ =\;& (1-s_k)\sum_{i=0}^{n-1}(1-s_i)(2i-2k+1) + \left[\sum_{i=0}^{k-1}(1-s_i)\right]^2 \end{aligned}$$

$$
\begin{aligned}
&- \left[\sum_{i=k}^{n-1}(1-s_i)\right]^2 \\
&= (1-s_k)\sum_{i=k}^{n-1}(1-s_i)(2i-2k+1) - \left[\sum_{i=k}^{n-1}(1-s_i)\right]^2 \\
&\quad + \left[\sum_{i=0}^{k-1}(1-s_i)\right]^2 - (1-s_k)\sum_{i=0}^{k-1}(1-s_i)(2k-2i-1).
\end{aligned}
$$

This establishes the lower bound on g_k.

To see that g_k is nonnegative, we proceed by letting $\sigma_i = s_{i+k-1},\ i = 1, \ldots, n-k$. Then by Lemma 4.6, we have that

$$
\begin{aligned}
&(1-s_k)\sum_{i=k}^{n-1}(1-s_i)(2i-2k+1) - \left[\sum_{i=k}^{n-1}(1-s_i)\right]^2 \\
&= (1-\sigma_1)\sum_{i=1}^{n-k}(1-\sigma_i)(2i-1) - \left[\sum_{i=1}^{n-k}(1-\sigma_i)\right]^2 \ \geq\ 0.
\end{aligned}
$$

On letting $\tau_i = s_{i-1},\ i = 1, \ldots, k+1$, we have from Lemma 4.3 that

$$
\begin{aligned}
&\left[\sum_{i=0}^{k-1}(1-s_i)\right]^2 - (1-s_k)\sum_{i=0}^{k-1}(1-s_i)(2k-2i-1) \\
&= \left[\sum_{i=1}^{k}(1-\tau_i)\right]^2 - (1-\tau_{k+1})\sum_{i=1}^{k}(1-\tau_i)(2k-2i+1) \ \geq\ 0.
\end{aligned}
$$

Note that if $g_k = 0$, then equality holds in Lemmas 4.6 and 4.3, from which we deduce that $s_1 = \cdots = s_{n-1} = 0$, and thus $a_1 = \cdots = a_{n-1} = 0$ and $a_n = 1$. The sufficiency of that condition to yield $g_k = 0$ is readily established. □

The foregoing lemmas and theorems yield the following lower bounds on the derivative of the elasticity of the Perron eigenvalue for the projection matrix in the size-classified population model.

THEOREM 4.8. *Let A be an irreducible stochastic population projection matrix as given in* (2.1). *Then, for $j = 0, 1, \ldots, n-1$,*

$$
\frac{\partial e_{1,j+1}}{\partial a_{1,j+1}} \geq w_1^3(1-s_{j+1})\sum_{i=0}^{j-1}(1-s_i)(2j-2i+1) \geq 0, \tag{4.13}
$$

with $\partial e_{1,j+1}/\partial a_{1,j+1} = 0$ if and only if $a_i = b_i = 0,\ 1 \leq i \leq n-1$ and $j+1 = n$; for

$j = 1, 2, \ldots, n-1$,

$$\frac{\partial e_{j+1,j}}{\partial a_{j+1,j}} \geq w_1^2 w_{j+1} \left\{ (1-s_j) \sum_{i=j}^{n-1} (1-s_i)(2i-2j+1) - \left[\sum_{i=j}^{n-1} (1-s_i) \right]^2 \right.$$

$$\left. + \left[\sum_{i=0}^{j-1} (1-s_i) \right]^2 - (1-s_j) \sum_{i=0}^{j-1} (1-s_i)(2j-2i-1) \right\} \tag{4.14}$$

$$\geq 0,$$

with $\partial e_{j+1,j}/\partial a_{j+1,j} = 0$ *if and only if* $a_1 = \cdots = a_{n-1} = 0$, $a_n = 1$, *and* $b_1 = \cdots = b_{n-1} = 0$.

We note that each of the lower bounds in (4.13) and (4.14) can be extended to provide a corresponding lower bound on the derivative of the elasticity of the Perron eigenvalue of a general, not necessarily stochastic, irreducible population projection matrix as given in (1.1). This extension is obtained via the transformation described in Remark 2.1 of producing a related stochastic matrix, applying the lower bounds arising from (4.13) and (4.14), and then appealing to (2.2).

REFERENCES

[1] K. Anstreicher and U. Rothblum, *Using Gauss–Jordan elimination to compute the index, generalized nullspace, and Drazin inverse*, Linear Algebra Appl., 85 (1987), pp. 221–239.

[2] A. Ben-Israel and T. N. Greville, *Generalized Inverses: Theory and Applications*, 2nd ed., Springer-Verlag, New York, 2003.

[3] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, Philadelphia, 1994.

[4] S. L. Campbell and C. D. Meyer, Jr., *Generalized Inverses of Linear Transformations*, Dover Publications, New York, 1991.

[5] H. Caswell, *Matrix Population Models: Construction, Analysis, and Interpretation*, 2nd ed., Sinauer, Sunderland, MA, 2001.

[6] L. Demetrius, *The sensitivity of population growth rate to perturbations in the life cycle components*, Math. Biosciences, 4 (1969), pp. 129–136.

[7] H. De Kroon, A. Plaisier, J. van Groenendael, and H. Caswell, *Elasticity: The relative contribution of demographic parameters to population growth rate*, Ecology, 65 (1986), pp. 1427–1431.

[8] E. Deutsch and M. Neumann, *Derivatives of the Perron root at an essentially nonnegative matrix and the group inverse of an M-matrix*, J. Math. Anal. Appl., 102 (1984), pp. 1–29.

[9] E. Deutsch and M. Neumann, *On the first and second order derivatives of the Perron vector*, Linear Algebra Appl., 71 (1985), pp. 57–76.

[10] R. Hartwig, *A method for calculating* A^d, Math. Japon., 26 (1981), pp. 37–43.

[11] S. J. Kirkland and M. Neumann, *Convexity and concavity of the Perron root and vector of Leslie matrices with applications to a population model*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1092–1107.

[12] S. J. Kirkland, M. Neumann, N. Ormes, and J. Xu, *On the elasticity of the Perron root of a nonnegative matrix*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 454–464.

[13] L. Lefkovitch, *The study of population growth in organisms grouped by stages*, Biometrics, 21 (1965), pp. 1–18.

[14] C. D. Meyer, *The role of the group generalized inverse in the theory of finite Markov chains*, SIAM Rev., 17 (1975), pp. 443–464.

[15] C. D. Meyer, *Sensitivity of the stationary distribution of a Markov chain*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 715–728.

[16] J. H. Pollard, *Mathematical Models for the Growth of Human Populations*, Cambridge University Press, Cambridge, UK, 1973.

[17] E. Seneta, *Nonnegative Matrices and Markov Chains*, 2nd ed., Springer Verlag, New York, 1981.

[18] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.
[19] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 1, pp. 186–214

V-CYCLE OPTIMAL CONVERGENCE FOR CERTAIN (MULTILEVEL) STRUCTURED LINEAR SYSTEMS*

ANTONIO ARICÒ†, MARCO DONATELLI‡, AND STEFANO SERRA-CAPIZZANO†

Abstract. In this paper we are interested in the solution by multigrid strategies of multilevel linear systems whose coefficient matrices belong to the circulant, Hartley, or τ algebras or to the Toeplitz class and are generated by (the Fourier expansion of) a nonnegative multivariate polynomial f. It is well known that these matrices are banded and have eigenvalues equally distributed as f, so they are ill-conditioned whenever f takes the zero value; they can even be singular and need a low-rank correction.

We prove the V-cycle multigrid iteration to have a convergence rate independent of the dimension even in presence of ill-conditioning. If the (multilevel) coefficient matrix has partial dimension n_r at level r, $r = 1, \ldots, d$, then the size of the algebraic system is $N(n) = \prod_{r=1}^{d} n_r$, $O(N(n))$ operations are required by our technique, and therefore the corresponding method is optimal.

Some numerical experiments concerning linear systems arising in applications, such as elliptic PDEs with mixed boundary conditions and image restoration problems, are considered and discussed.

Key words. circulant, Hartley, and τ algebra, Toeplitz class, two-grid and multigrid iterations, multi-iterative methods, multilevel matrices

AMS subject classifications. 65F10, 65F15, 15A12

DOI. 10.1137/S0895479803421987

1. Introduction.

Let $f(x)$, $x = (x_1, \ldots, x_d)$, be a continuous function on the d-dimensional Euclidean space $\mathbb{R}^d$, and let $\langle \cdot | \cdot \rangle$ denote the usual scalar product between vectors. Henceforth, we suppose that f has period 2π with respect to each variable and is real valued, so the Fourier coefficients of f,

$$a_j = \frac{1}{(2\pi)^d} \int_{[-\pi,\pi]^d} f(x) e^{-\mathbf{i}\langle j|x\rangle}\, dx, \qquad \mathbf{i}^2 = -1, \qquad j = (j_1, \ldots, j_d) \in \mathbb{Z}^d, \tag{1.1}$$

enjoy the relation $a_{-j} = \bar{a}_j$ for every $j \in \mathbb{Z}^d$. From the coefficients a_j one can build [35] the sequence $\{T_n(f)\}$, $n = (n_1, \ldots, n_d) \in \mathbb{N}^d$, of multilevel Toeplitz matrices of size $N(n) = \prod_{r=1}^{d} n_r$. Every matrix $T_n(f)$ is explicitly written as

$$T_n(f) = \sum_{|j| \leqslant n-e} a_j J_n^{[j]} = \sum_{|j_1| \leqslant n_1 - 1} \cdots \sum_{|j_d| \leqslant n_d - 1} a_{(j_1,\ldots,j_d)} J_{n_1}^{[j_1]} \otimes \cdots \otimes J_{n_d}^{[j_d]}.$$

Here $\otimes$ denotes the usual tensor product, so that $A \otimes B$ is the block matrix $[a_{ij}B]_{ij}$, $e = (1, \ldots, 1) \in \mathbb{N}^d$ and the relations between two multi-indices (as $|j| \leqslant n - e$) should be intended componentwise. If n and j are integer numbers, then $J_n^{[j]} \in \mathbb{R}^{n \times n}$ is the matrix whose entry (s,t) equals 1 if $s - t = j$ and is 0 elsewhere; in the case where n and j are multi-indices, the symbol $J_n^{[j]}$ denotes the tensor product of all the $J_{n_r}^{[j_r]}$ for $r = 1, \ldots, d$. From the identity $a_{-j} = \bar{a}_j$ for every j, it follows that the matrices

*Received by the editors January 31, 2003; accepted for publication (in revised form) by L. Reichel September 19, 2003; published electronically August 27, 2004.
http://www.siam.org/journals/simax/26-1/42198.html

†Dipartimento di Matematica "Felice Casorati," Università di Pavia, Via Ferrata 1, 27100, Pavia, Italy (arico@dimat.unipv.it, arico@dm.unipi.it).

‡Dipartimento di Chimica, Fisica e Matematica, Università dell'Insubria–Sede di Como, Via Valleggio 11, 22100 Como, Italy (marco.donatelli@uninsubria.it, stefano.serrac@uninsubria.it, serra@mail.dm.unipi.it).

$T_n(f)$ are Hermitian for every n. It is clear that, if f is a trigonometric polynomial of degree $c = (c_1, \ldots, c_d)$, then the Fourier coefficient a_j equals zero when $|j| \leqslant c$ is not satisfied; in that case the corresponding matrix $T_n(f)$ shows a d-level structure with bandwidth c_r at level $r \in \{1, \ldots, d\}$.

To the same coefficients a_j in (1.1) we can also associate matrices belonging to well-known trigonometric (multilevel) algebras. For instance, the d-level circulant related to f is defined as

$$\mathcal{C}_n(f) = \sum_{|j| \leqslant n-e} a_j Z_n^{[j]} = \sum_{|j_1| \leqslant n_1 - 1} \cdots \sum_{|j_d| \leqslant n_d - 1} a_{(j_1,\ldots,j_d)} Z_{n_1}^{j_1} \otimes \cdots \otimes Z_{n_d}^{j_d},$$

where $Z_n = J_n^{[-1]} + \mathbf{e}_n \mathbf{e}_1^t$ if n is a scalar and $\mathbf{e}_j$ denotes the jth vector of the canonical basis. Analogously to the Toeplitz case, if n and j are multi-indices, then $Z_n^{[j]}$ represents the tensor product of all the $Z_{n_r}^{j_r}$ for $r = 1, \ldots, d$. If f is even (with regard to each variable x_r separately) we have $a_j = a_{-j} \in \mathbb{R}$, i.e., $T_n(f)$ is real and symmetric. In that case an interesting matrix algebra approximation is provided by the τ algebra [3]. More specifically we define

$$\tau_n(f) = \sum_{0 \leqslant j \leqslant n-e} b_j H_n^{[j]} = \sum_{0 \leqslant j_1 \leqslant n_1 - 1} \cdots \sum_{0 \leqslant j_d \leqslant n_d - 1} b_{(j_1,\ldots,j_d)} H_{n_1}^{j_1} \otimes \cdots \otimes H_{n_d}^{j_d},$$

where $H_n = J_n^{[1]} + J_n^{[-1]}$, the matrix $J_n^{[j]}$ is defined as before, and the coefficients b_j can be uniquely determined by the coefficients a_j through an invertible triangular linear system (see [24]). A further characterization of the τ algebra is obtained by observing that every matrix of the class can be written as a Toeplitz plus Hankel matrix (a Hankel matrix is constant along the antidiagonals): more precisely, we have

$$\tau_n(f) = T_n(f) - H_n(f), \tag{1.2}$$

where $H_n(f)$ is the centrosymmetric Hankel matrix generated by f. A Hankel matrix is such that its entries are constant along any lower-left–upper-right diagonal: with the same notations we have

$$H_n(f) = \sum_{2e \leqslant j \leqslant n-e} a_j K_n^{[j]} = \sum_{2 \leqslant |j_1| \leqslant n_1 - 1} \cdots \sum_{2 \leqslant |j_d| \leqslant n_d - 1} a_{(j_1,\ldots,j_d)} K_{n_1}^{[j_1]} \otimes \cdots \otimes K_{n_d}^{[j_d]}, \tag{1.3}$$

where, in the unilevel case, $K_n^{[j]}$ denotes the matrix of order n whose entry (s,t) equals 1 if $s + t = j \bmod 2(n-1)$ and equals zero otherwise: the multilevel version of $K_n^{[j]}$ is now defined via (1.3).

A third class of matrices which form an algebra and is of interest in applications is represented by the Hartley matrices [4]. Unlike circulants and τ matrices, the Hartley class does not have a generator, but it can be described by using circulant matrices. In actuality, every matrix belonging to this class can be expressed as the sum of two independent matrices, the first being symmetric and circulant, the second being the product of a special permutation matrix J by a skewcirculant matrix. More precisely, for a Hartley matrix generated by a unilevel function f we set $J_{1,1} = J_{s,n+2-s} = 1$, $s = 2, \ldots, n$, and

$$\mathcal{H}_n(f) = \mathcal{C}_n(f_{\text{even}}) + J\mathcal{C}_n(f_{\text{odd}}),$$

where $f_{\text{even}}(x) = (f(x) + f(-x))/2$ and $f_{\text{odd}}(x) = (f(x) - f(-x))/2$. In this way the first column of $\mathcal{C}_n(f_{\text{even}})$ has α_j coefficients such that $\alpha_j = \alpha_{n-j} \in \mathbb{R}$, $j = 1, \ldots, n-1$, and the first column of $\mathcal{C}_n(f_{\text{odd}})$ has coefficients $\beta_j = -\beta_{n-j} \in \mathbb{R}$, $j = 1, \ldots, n-1$, $\beta_0 = 0$, where $\alpha_0 = a_0$ and $(\alpha_j - \mathbf{i}\beta_j)/2 = a_j$ for $|j| \geqslant 1$. We note that its multilevel version amounts to performing the same even/odd splitting of f with respect to each variable separately.

Since circulants, τ and Hartley matrices are algebras, they are all simultaneously diagonalized by a given transform. In our case the involved transforms are all unitary (or real unitary, i.e., orthogonal) and therefore these algebras are constituted by normal matrices. More precisely the three classes can be formally defined as follows:

$$\mathcal{G}(Q_n) = \left\{ Q_n \cdot \text{Diag}\,(\mathbf{d}) \cdot Q_n^{-1} \mid \mathbf{d} \in \mathbb{C}^n \right\} = \left\{ Q_n \cdot \text{Diag}\,(\mathbf{d}) \cdot Q_n^{H} \mid \mathbf{d} \in \mathbb{C}^n \right\},$$

where the related transforms Q_n (and some other information, such as the grid points $w_i^{[n]}$, the $\mathcal{I}_n$ index range to which i belongs, and the name of the class $\mathcal{C}, \tau, \mathcal{H}$ generically denoted by $\mathcal{A}$) are listed in the subsequent Table 1.1.

TABLE 1.1
Basics on our algebras: the unilevel case.

	$\mathcal{A}$	$\mathcal{I}_n$	$\mathbf{w}^{[n]}$	Q_n
Circulants	$\mathcal{C}$	$0, \ldots, n-1$	$w_i^{[n]} = \frac{2\pi i}{n}$	$F_n = \frac{1}{\sqrt{n}} \left[e^{\mathbf{i} j w_i^{[n]}} \right]_{i,j=0}^{n-1}$
Hartley	$\mathcal{H}$	$0, \ldots, n-1$	$w_i^{[n]} = \frac{2\pi i}{n}$	$\text{Re}(F_n) + \text{Im}(F_n)$
Tau	τ	$1, \ldots, n$	$w_i^{[n]} = \frac{\pi i}{n+1}$	$\sqrt{\frac{2}{n+1}} \left[\sin\left(j w_i^{[n]} \right) \right]_{i,j=1}^{n}$

Once again, whenever n is a d-index we define Q_n the matrix of size $N(n)$ as $Q_{n_1} \otimes \cdots \otimes Q_{n_d}$. The matrices $\mathcal{C}_n(f)$, $\tau_n(f)$, and $\mathcal{H}_n(f)$ can be written (in order to provide a uniform approach) as

$$\mathcal{A}_n(f) = Q_n \cdot \text{Diag}\left(f\left(\mathbf{w}^{[n]} \right) \right) \cdot Q_n^{H}, \tag{1.4}$$

where $\mathcal{A} \in \{\mathcal{C}, \mathcal{H}, \tau\}$, f is a polynomial of degree less than n, and the vectors $\mathbf{w}^{[n]}$ are defined in the fourth column of Table 1.1 for scalar n and $\mathbf{w}^{[n]} = \mathbf{w}^{[n_1]} \times \cdots \times \mathbf{w}^{[n_d]}$ if n is a d-index. For instance, in the circulant case we observe $Q_n = F_n$ and we write $\mathcal{A}_n(f) = \mathcal{C}_n(f) = F_n \cdot \text{Diag}\left(f(\mathbf{w}^{[n]}) \right) \cdot F_n^H$.

It is immediate to see that $\mathcal{C}_n(f)$, $\tau_n(f)$, and $\mathcal{H}_n(f)$ are definitely ill-conditioned if f has zeros in its basic definition set $[-\pi, \pi]^d$ (they are singular if the zeros contain a grid point). It is interesting to recall that $x = 0$ is always a grid point for the circulants and the Hartley matrices so that $\mathcal{C}_n(f)$ and $\mathcal{H}_n(f)$ are singular if these matrices arises from the discretization of constant coefficients differential operators: in that case it is known that $x = 0$ is a zero of the symbol and its order is associated to the maximal order of the involved derivatives (see, e.g., [30]).

In such a case, setting $\mathbf{e} = \sum_{j=1}^{N(n)} \mathbf{e}_j$, the classical Strang circulant preconditioner (see, e.g., [9]) is replaced by its modified (or stabilized) version (see, e.g., [34]):

$$\widetilde{\mathcal{C}}_n(f) = \mathcal{C}_n(f) + \left(\min_{\|j\|_\infty = 1} f\left(\mathbf{w}_j^{[n]} \right) \right) \frac{\mathbf{e}\mathbf{e}^t}{N(n)}. \tag{1.5}$$

Of course the same approach can be followed in the case of the Hartley algebra.

In this paper we are interested in the solution of linear systems with matrices of the form $\mathcal{A}_n(f)$ for $\mathcal{A} \in \{\mathcal{C}, \mathcal{H}, \tau, T\}$ and f trigonometric polynomial. More specifically we are interested in (iterative) methods that show the best possible asymptotic complexity. In this respect we define a formal notion of optimality of an iterative method for a sequence of linear systems of increasing dimensions.

DEFINITION 1.1. *Given a sequence of linear systems of increasing dimensions* $\{A_n \mathbf{x}_n = \mathbf{b}_n\}$*, we write that an iterative method is* optimal *if*

1. *the arithmetic cost of every iteration is at most proportional to the complexity of a matrix vector product with matrix* A_n*,*
2. *the number of iterations for reaching the solution within a fixed accuracy can be bounded from above by a constant independent of* n*.*

Such a method would be interesting in the case of the considered matrix algebras $\mathcal{A} \in \{\mathcal{C}, \mathcal{H}, \tau\}$ since the cost by direct methods using fast transforms is $O(N(n) \log N(n))$ while an optimal technique would require just $O(N(n))$ operations: we recall that this kind of matrix algebra linear systems are widely used as preconditioners for more complicated problems (dense Toeplitz, differential problems discretizations etc. [13, 25, 30]) or directly arise in the discretization of image restoration problems with shift-invariant kernel and suitable boundary conditions (see [20, 29]).

In the case of Toeplitz systems the improvement would be much more striking. For instance, in the multilevel Toeplitz setting the fast direct techniques are expensive because they are unable to exploit the Toeplitzness at each level. Concerning the preconditioned conjugate gradient (PCG) method, the matrix algebra preconditioners lead to optimal solvers only in the unilevel case (see, e.g., [9]). Unfortunately, for multilevel problems the optimal preconditioning by matrix algebras is simply impossible in general as proved by the last author and Tyrtyshnikov [32] (see also [21, 22, 26, 33]). More precisely the number of iterations is an unbounded function as n and it is of the order of $O([N(n)]^{\frac{d-1}{d}})$, which is very unsatisfactory if d is large.

On the other hand, by using band Toeplitz preconditioners (see, e.g., [13]), it is possible to reduce the computation with dense Toeplitz systems to the case of Toeplitz linear systems whose coefficient matrices are generated by nonnegative polynomials. Therefore it is of special interest to be able to solve in optimal time (i.e., computational effort linear with respect to the size of the algebraic problem) linear systems whose coefficient matrix is of the form $T_n(f)$ with nonnegative polynomial f and our proposal is the multigrid technique. We will give a formal proof of optimality of the V-cycle multigrid iteration (MGM) in the matrix algebra case while in the Toeplitz case this optimal behavior is demonstrated only by numerical experiments (for a formal proof of optimal convergence rate, i.e., independent of n, related to the two-grid method refer to [27]): our hope is that the theoretical tools introduced in this paper for the matrix algebra case could be used for proving the V-cycle optimality in the Toeplitz context as well. We stress that the proof technique introduced in this paper seems to be new compared with the classical approaches used in the PDEs context (see, e.g., the beautiful review [37]). Indeed our tools are totally matrix oriented so that there is no differential interpretation in the general case: for instance when the symbol has a zero close to π, then the smoother (i.e., the iteration satisfying the smoothing property according to Ruge and Stüben [23]) does not make the error smooth; i.e., it reduces the components in the low frequencies and it does not reduce the components in the high frequencies.

Finally, we mention that our technique can be easily extended with the same

linear arithmetic cost to linear systems with coefficient matrices given by $\mathcal{A}_n(f) + \sum_{t=1}^{s} \vartheta_t \cdot q_{i_t}^{[n]} \big(q_{i_t}^{[n]}\big)^H = \mathcal{A}_n(f + \sum_{t=1}^{s} \vartheta_t \chi_{w_{i_t}^{[n]}+2\pi\mathbb{Z}})$, where the vector $q_i^{[n]}$ is the ith column of Q_n, $\mathcal{A} \in \{\mathcal{C}, \mathcal{H}, \tau\}$, and χ_S denotes the characteristic function of a given set S. We recall that these examples of coefficient matrices are a generalization of the stabilized Strang preconditioner displayed in (1.5).

The paper is organized as follows. In section 2 we first introduce the multigrid procedure by reporting the basic convergence results by Ruge and Stüben [23]; then we describe our choices for the smoothing and prolongation operators. In section 3 we show that "level independency" property is not sufficient to reach the optimality, while in section 4 we prove the optimal convergence rate of our multigrid in the unilevel case. Section 5 is devoted to the multilevel case, while in section 6 we generalize our V-cycle algorithm to multilevel Toeplitz matrices. Section 7 contains wide numerical experimentation that confirms the theoretical analysis, and section 8 is devoted to concluding remarks, open problems and future work.

2. The multigrid procedure. Let $A \in \mathbb{C}^{n\times n}$ be a Hermitian positive definite matrix, $\mathbf{b} \in \mathbb{C}^n$, m integer with $0 < m < n$. Fix integers $n_0 = n > n_1 > n_2 > \cdots > n_m > 0$, take $P_{i+1}^i \in \mathbb{C}^{n_{i+1}\times n_i}$ full-rank matrices, and consider a class $\mathcal{R}_i$ of iterative methods for n_i-dimensional linear systems. The related V-cycle method (see [5, 17]) produces the sequence $\{\mathbf{x}^{(k)}\}_{k\in\mathbb{N}}$ according to the rule $\mathbf{x}^{(k+1)} = \mathcal{MGM}(0, \mathbf{x}^{(k)}, \mathbf{b})$, with $\mathcal{MGM}$ recursively defined as follows:

$$\mathbf{x}_i^{(\mathrm{out})} := \mathcal{MGM}(i, \mathbf{x}_i^{(\mathrm{in})}, \mathbf{b}_i)$$

(2.1)

```
If (i = m) Then Solve(A_m x_m^(out) = b_m)
Else 1 | r_i     := A_i x_i^(in) − b_i
     2 | b_{i+1} := P_{i+1}^i r_i
     3 | A_{i+1} := P_{i+1}^i A_i (P_{i+1}^i)^H
     4 | y_{i+1} := MGM(i + 1, 0_{n_{i+1}}, b_{i+1})
     5 | x_i^(int) := x_i^(in) − (P_{i+1}^i)^H y_{i+1}
     6 | x_i^(out) := R_i^ν (x_i^(int)).
```

Step 1 calculates the residual of the proposed solution; steps 2, 3, 4, and 5 define the *recursive coarse grid correction* by projection (2) of the residual, sub-grid correction (3, 4) and interpolation (5), while step 6 performs some (ν) iterations of a "post-smoother."

By using the MGM as an iterative technique, at the kth iteration, we obtain the linear systems $A_i\mathbf{x}_i^{(k)} = \mathbf{b}_i^{(k)}$, $i = 0, \ldots, m$, where the matrices $A_i \in \mathbb{C}^{n_i\times n_i}$ are Hermitian positive definite. Only the last is solved exactly while all the others are recursively managed by reduction to low-level system and smoothing. $\mathcal{R}_i$ are most of the time one-point methods (see [23]) with prescribed linear part $R_i \in \mathbb{C}^{n_i\times n_i}$ i.e.,

$$\mathcal{R}_i(\mathbf{x}_i) = R_i\mathbf{x}_i + (I_{n_i} - R_i)A_i^{-1}\mathbf{b}_i^{(k)}, \quad \mathbf{x}_i \in \mathbb{C}^{n_i}, \quad i = 0, \ldots, m-1. \tag{2.2}$$

If we define the multigrid iteration matrix of level i as MGM_i,

$$\begin{cases} MGM_m = O_{n_m\times n_m}, \\ MGM_i = R_i^\nu \cdot \Big[I_{n_i} - \big(P_{i+1}^i\big)^H \big(I_{n_{i+1}} - MGM_{i+1}\big) A_{i+1}^{-1} P_{i+1}^i A_i\Big], \quad i = m-1, \ldots, 0, \end{cases} \tag{2.3}$$

it holds that $\mathbf{x}_i^{(\text{out})} = MGM_i\mathbf{x}_i^{(\text{in})} + (I_{n_i} - MGM_i)A_i^{-1}\mathbf{b}_i$, so in the finer grid we have $\mathbf{x}^{(\text{k}+1)} = MGM_0\mathbf{x}^{(\text{k})} + (I_{n_0} - MGM_0)A_i^{-1}\mathbf{b}$, and MGM_i depends on i but not on any $\mathbf{x}_i^{(k)}$ nor on $\mathbf{b}_i^{(k)}$. The algorithm has essentially two degrees of indetermination:

1. choice of the projectors P_{i+1}^i, $i = 0, \ldots, m-1$;
2. choice of the smoothers $\mathcal{R}_i$, $i = 0, \ldots, m-1$.

The choice of the projectors P_{i+1}^i and the calculation of the matrices A_i are performed before the beginning of the V-cycle procedure (precomputing phase).

Of course m stands for the number of subgrids in the algorithm. We also refer the choice $m = 1$ as two-grid method (TGM), so we define the TGM linear action as

$$TGM_0 = R_0^\nu \cdot \Big[I_{n_0} - \big(P_1^0\big)^H A_1^{-1} P_1^0 A_0\Big]. \tag{2.4}$$

The term in square brackets in (2.4) is defined as *exact coarse grid correction* (CGC_0). It can be defined of course on each grid of the V-cycle algorithm, and hence we agree to write

$$CGC_i = I_{n_i} - \big(P_{i+1}^i\big)^H A_{i+1}^{-1} P_{i+1}^i A_i, \qquad i = 0, \ldots, m-1, \tag{2.5}$$

and

$$TGM_i = R_i^\nu \cdot CGC_i, \qquad i = 0, \ldots, m-1. \tag{2.6}$$

Specific TGMs and V-cycles have been devised for the τ multilevel algebra [15, 16], while for multilevel circulants they have been studied in [31].

2.1. Convergence related theorems. Here we recall two theorems [23] concerning the convergence of multigrid iterations. The first one is related to the easier TGM algorithm; the other refers to the complete (i.e., with $m > 1$) multigrid procedure. For the sake of simplicity, in both the theorems we will assume just one application of the smoother, i.e., $\nu = 1$. By $\|\cdot\|_2$ we denote the Euclidean norm on $\mathbb{C}^n$ and the associated induced norm on $\mathbb{C}^{n\times n}$; if X is positive definite we also denote $\|\cdot\|_X = \|X^{1/2}\cdot\|_2$, and whenever X and Y are both Hermitian matrices the notation $X \geqslant Y$ means that $X - Y$ is positive semidefinite.

THEOREM 2.1 (TGM convergence [23]). *Let n_0, n_1 be integers such that $n_0 > n_1 > 0$ and let $A \in \mathbb{C}^{n_0\times n_0}$ be a positive definite Hermitian matrix, $\mathbf{b} \in \mathbb{C}^{n_0}$, and also let $\mathcal{R}_0$ be defined as in* (2.2). *Fix $P_1^0 \in \mathbb{C}^{n_1\times n_0}$ full-rank matrix and let $D = \text{Diag}[a_{ii}]_{i=1}^{n_0}$ be the main diagonal of A. Suppose that $\alpha > 0$ exists such that*

$$\|R_0\mathbf{x}\|_A^2 \leqslant \|\mathbf{x}\|_A^2 - \alpha\,\|\mathbf{x}\|_{AD^{-1}A}^2 \qquad \forall\,\mathbf{x} \in \mathbb{C}^{n_0}. \tag{2.7a}$$

Then for each $\gamma > 0$ such that

$$\min_{\mathbf{y}\in\mathbb{C}^{n_1}} \|\mathbf{x} - (P_1^0)^H\mathbf{y}\|_D^2 \leqslant \gamma\,\|\mathbf{x}\|_A^2 \qquad \forall\,\mathbf{x} \in \mathbb{C}^{n_0} \tag{2.7b}$$

it holds that $\alpha \leqslant \gamma$ and

$$\|TGM_0\|_A \leqslant \sqrt{1 - \alpha/\gamma} < 1. \tag{2.8}$$

From Theorem 2.1, it follows that $\{\mathbf{x}^{(k)}\}_k$ converges to the solution of $A\mathbf{x} = \mathbf{b}$. Furthermore, when α and γ are independent of n, the sequence $\{\mathbf{x}^{(k)}\}_k$ converges with (at least) a constant error reduction by the factor $\sqrt{1 - \alpha/\gamma}$ independent of the

dimension n of the system: therefore the corresponding TGM has optimal convergence rate (i.e., it satisfies the second item in Definition 1.1).

Remark 2.2. Theorem 2.1 still holds if $D \in \mathbb{C}^{n_0 \times n_0}$ is replaced by any Hermitian positive definite matrix X ($X = I_{n_0}$ could be a suitable choice): it is enough to repeat verbatim the proof of Theorem 5.2 in [23] with X in place of D.

THEOREM 2.3 (MGM convergence [23]). *Let m, n be integers satisfying $0 < m < n$ and suppose that $A \in \mathbb{C}^{n \times n}$ is a positive definite Hermitian matrix and $\mathbf{b} \in \mathbb{C}^n$; given now a sequence of $m+1$ positive integers $n = n_0 > n_1 > \cdots > n_m$, let $P_{i+1}^i \in \mathbb{C}^{n_{i+1} \times n_i}$ be full-rank matrices for each $i = 0, \ldots, m-1$. Define $A_0 = A$ and choose a class of iterative methods $\mathcal{R}_i$ as in* (2.2)*. If there exists a real positive number δ satisfying*

$$\|R_i\,\mathbf{x}\|_{A_i}^2 \leqslant \|\mathbf{x}\|_{A_i}^2 - \delta\,\|CGC_i\,\mathbf{x}\|_{A_i}^2 \qquad \forall\,\mathbf{x} \in \mathbb{C}^{n_i} \tag{2.9}$$

for every $i = 0, \ldots, m-1$, then it holds $\delta \leqslant 1$ and

$$\|MGM_0\|_A \leqslant \sqrt{1-\delta} < 1. \tag{2.10}$$

As in the case of the TGM, here also the sequence $\{\mathbf{x}^{(k)}\}_k$ converges to the solution of $A\mathbf{x} = \mathbf{b}$ and when δ is independent of n it converges with at least a constant error reduction not depending on the dimension of the system and, at most, $\lceil 2\delta^{-1} \ln(\varepsilon^{-1}) \rceil$ iterations are needed to reduce the error by a factor $\varepsilon > 0$.

We observe that inequality (2.9) is easily guaranteed by the following:

$$\|R_i\,\mathbf{x}\|_{A_i}^2 \leqslant \|\mathbf{x}\|_{A_i}^2 - \alpha_i\,\|\mathbf{x}\|_{A_i^2}^2 \quad (\alpha_i > 0) \qquad \forall\,\mathbf{x} \in \mathbb{C}^{n_i}, \tag{2.11a}$$

$$\|CGC_i\,\mathbf{x}\|_{A_i}^2 \leqslant \beta_i\,\|\mathbf{x}\|_{A_i^2}^2 \qquad \forall\,\mathbf{x} \in \mathbb{C}^{n_i}. \tag{2.11b}$$

If $\delta \leqslant \alpha_i/\beta_i$, then (2.9) holds for every $i = 0, \ldots, m-1$ with the choice of $\delta = \min_{0 \leqslant i \leqslant m-1}\{\alpha_i/\beta_i\}$. We refer to (2.11a) as the *smoothing property* and to (2.11b) as the *approximation property* (see [23, 37]). The approximation property depends exclusively on the choice of projectors (i.e., P_{i+1}^i) but not on smoothers, whereas smoothing property is not related to P_{i+1}^i. The separate study of these two properties allows us to cope with the difficult part of the procedure (the verification of condition (2.11b)) involving the projectors but not depending on the smoothers. Notice that the direct verification of (2.9) is in principle much more intricate due to the simultaneous presence of the projectors and of the smoothers in the inequalities.

Remark 2.4. The MGM smoothing property (2.11a) is nothing more than the TGM smoothing property (2.7a) with D substituted by I, in accordance with Remark 2.2.

In such a situation, optimality is reached if δ is independent from both n and m, i.e., it suffices to show that a constant value δ exists such that $0 < \delta \leqslant \min_i\{\alpha_i/\beta_i\}$ is fulfilled for every possible choice of n and m. In this way, the number of iterations required keeps being uniformly bounded by a constant irrespective of the dimension of the problem. What is more, since each iteration has a computational cost proportional to matrix-vector product, Definition 1.1 states that such a kind of MGM is *optimal*.

2.2. MGM for matrix algebras. We analyze a special instance of the MGM (2.1), introduced in [15, 16, 31], where the smoother is the relaxed Richardson iteration, namely $R_i = I_{n_i} - \omega_i A_i$ (ω_i is relaxing parameter), and on each step we essentially halve the dimension ($n_{i+1} = \frac{n_i}{2}$ for circulants and Hartley and $n_{i+1} = \frac{n_i - 1}{2}$ for τ matrices).

Dealing with circulants or Hartley matrices we start from dimension $n_0 = 2^{k_0}$ and define the subgrid dimensions as $n_i = 2^{k_0-i}$, while in τ algebra we start with $n_0 = 2^{k_0} - 1$ and define $n_i = 2^{k_0-i} - 1$. The cutting operator is defined by $K^i_{i+1} : \mathbb{C}^{n_i} \longrightarrow \mathbb{C}^{n_{i+1}}$ and it selects even index components (we recall that the index range is $\{0, \dots, n_i - 1\}$ in the circulant and Hartley algebras, while it is $\{1, \dots, n_i\}$ in the τ algebra):

(2.12)

Circulant & Hartley algebra	τ algebra
$n_i = 2^{k_0-i}$	$n_i = 2^{k_0-i} - 1$
$K^i_{i+1} = \begin{bmatrix} 1\ 0 & & & \\ & 1\ 0 & & \\ & & \ddots\ \ddots & \\ & & & 1\ 0 \end{bmatrix}_{n_{i+1}\times n_i}$	$K^i_{i+1} = \begin{bmatrix} 0\ 1\ 0 & & & \\ & 0\ 1\ 0 & & \\ & & \ddots\ \ddots\ \ddots & \\ & & & 0\ 1\ 0 \end{bmatrix}_{n_{i+1}\times n_i}$

We have defined the projector in the form $P^i_{i+1} = K^i_{i+1} \cdot \mathcal{A}_{n_i}(p_i)$ while p_i is a real valued polynomial which will be chosen in section 4.2 in order to satisfy the approximation property (2.11b). Our choices on K^i_{i+1} brings (see [31]) to $K^i_{i+1} Q_{n_i} = \left[Q_{n_{i+1}} \mid Q_{n_{i+1}}\right]$ in the case of the circulant and Hartley algebras, while dealing with τ matrices we have $K^i_{i+1} Q_{n_i} = \left[Q_{n_{i+1}} \,|\mathbf{0}_{n_i+1}| - J_{n_{i+1}} Q_{n_{i+1}}\right]$ with $[J_n]_{h,k}$ equals 1 if $h + k = n + 1$ and 0 if not. These two equalities play a basic role in maintaining the matrix algebra structure on subgrids and represent the keystone for proving the following proposition.

PROPOSITION 2.5 (see [27, 31]). *Let k_0, m be integers such that $0 < m < k_0$, f_0 and p_i, $i = 0, \dots, m-1$, be real 2π-periodic functions (also even in the τ case), $P^i_{i+1} = K^i_{i+1} \cdot \mathcal{A}_{n_i}(p_i)$ with $\mathcal{A} \in \{\mathcal{C}, \mathcal{H}, \tau\}$ as in (1.4). Also define $A_0 = \mathcal{A}_{n_0}(f_0)$ and $A_{i+1} = P^i_{i+1} A_i (P^i_{i+1})^H$ for $i = 0, \dots, m-1$. Then it holds that $A_{i+1} = \mathcal{A}_{n_{i+1}}(f_{i+1})$, where*

$$f_{i+1}(x) = \frac{1}{2}\left[\left(p_i^2 f_i\right)\left(\frac{x}{2}\right) + \left(p_i^2 f_i\right)\left(\pi + \frac{x}{2}\right)\right], \qquad i = 0, \dots, m-1. \tag{2.13}$$

Moreover each projector P^i_{i+1} is full-rank if $p_i^2(x) + p_i^2(\pi + x) > 0$ holds true for every x.

Proposition 2.5 is basic for our purposes because it allows one to relate the functions f_i to the matrices A_i in the V-cycle procedure (2.1). Furthermore, we observe that

$$h(x) = \sum_{j=k_1}^{k_2} a_j e^{\mathbf{i}jx} \qquad \Rightarrow \qquad h\Big(\frac{x}{2}\Big) + h\Big(\pi + \frac{x}{2}\Big) = 2 \sum_{j=\lceil \frac{k_1}{2} \rceil}^{\lfloor \frac{k_2}{2} \rfloor} a_{2j} e^{\mathbf{i}jx} \tag{2.14}$$

represents a fundamental simplification in checking convergence and in evaluating the computational costs. By defining $\mathbb{R}_k[x] = \left\{ \sum_{|j|\leqslant k} a_j e^{\mathbf{i}jx} \mid a_j = \bar{a}_{-j} \in \mathbb{C} \right\}$, and by assuming $f(x) \in \mathbb{R}_{T_0}[x]$, $p_i(x) \in \mathbb{R}_{q_i}[x]$, we have $f_i \in \mathbb{R}_{T_i}$ with $T_{i+1} = q_i + \left\lfloor \frac{T_i}{2} \right\rfloor$, and, by induction, we deduce $T_i \leqslant \max\{T_0; 2q_j - 1 : 1 \leqslant j \leqslant i\}$. Consequently the bandwidth of $\mathcal{A}_i$ is uniformly bounded if there exists a constant T such that $T_0, q_i \leqslant T$ for every i. Furthermore, if $q_i = q$ holds for every i, then $T_i \uparrow 2q - 1$ (monotonic nondecreasing convergence) if $T_0 \leqslant 2q - 1$, $T_i \downarrow 2q$ (monotonic nonincreasing convergence) otherwise.

The subsequent theorem has been proven in [27, 31], where it has been used for proving the TGM optimality if a good choice of p_0 is performed.

THEOREM 2.6 (see [27, 31]). *Let $A_{n_0} = \mathcal{A}_n(f_0)$ with $\mathcal{A} \in \{\mathcal{C}, \mathcal{H}, \tau\}$, f_0 be nonnegative, 2π-periodic (even in the τ case), and let $P_1^0 = K_1^0 \cdot \mathcal{A}_{n_0}(p_0)$, with p_0 trigonometric polynomial (also even in τ case) such that $f_0(x^0) = 0$ implies*

$$\lim_{x \to x^0} \frac{p_0^2(\pi + x)}{f_0(x)} < +\infty, \tag{2.15a}$$

$$p_0^2(x) + p_0^2(\pi + x) > 0 \qquad \forall\, x. \tag{2.15b}$$

Then inequality (2.7b) *in Theorem* 2.1 *is satisfied.*

In what follows we will need a stronger version of (2.15a) to prove the V-cycle optimality.

3. Level independency does not imply multigrid optimality. An informal but dangerous (as we will see) way of defining the MGM is as recursive application of TGM iterations. In particular, if the convergence rate $\sqrt{1 - \alpha/\gamma}$ defined in (2.8) is independent of the recursion level, we have a property known in literature as "level independency" [7].

DEFINITION 3.1. *Let m, n be integers satisfying $0 < m < n$ and let us suppose that we solve a system of dimension n with MGM. Then we have* level independency *if the method TGM_i induced on each level satisfies*

$$\|TGM_i\|_{A_i} \leqslant c < 1, \qquad i = 0, \ldots, m-1,$$

with c pure constant independent of n and m.

In some recent works, the level independency was indicated as a way for obtaining the V-cycle optimality (see, e.g., [7]). Actually, we will prove that *the level independency is necessary but not sufficient for the MGM optimality.* To explain this fact intuitively, we observe that to consider the MGM as a recursive TGM application is equivalent to having the exact knowledge of the error at each level, since the TGM directly solves the system at the lower level. Indeed, for applying the TGM recursively, we must only decide if the recursive call should be placed before or after the direct resolution of the lower level system. It follows that in the first case we project the problem at the lower level as for the MGM, but when we interpolate the solution (the error) at each level this is exactly known at the lower level and it does not derive from previous interpolation as for the MGM. In the second case we know exactly the error that we project at each level, while for the MGM this derives from previous projections. On the other side, MGM replaces the direct solution of the system with the recursive call, obtaining a more approximate procedure with respect to the recursive TGM application. Therefore, the level independency is a necessary but not sufficient condition for the MGM optimality.

Now we report a whole class of counterexamples to enhance the previous informal description.

PROPOSITION 3.2. *Let $A = \tau_{n_0}(f_0)$, $P_{i+1}^i = K_{i+1}^i \tau_{n_i}(p)$, and*

$$f_0(x) = (1 - \cos(x))^q, \qquad q \in \mathbb{N},$$
$$p(x) = \mu^{\lceil \frac{q}{2} \rceil} (1 + \cos(x))^{\lceil \frac{q}{2} \rceil}, \qquad \mu \in \mathbb{R},\ \mu \neq 0,$$

for $i = 0, \ldots, m-1$. Then the level independency property holds for the MGM applied to the system $A\mathbf{x} = \mathbf{b}$, $\mathbf{x}, \mathbf{b} \in \mathbb{C}^{n_0}$, where P_{i+1}^i is the projector at the level i.

Proof. Following Definition 3.1 we must prove that $\|TGM_i\|_{A_i} \leqslant c < 1$ with c absolute constant and, to this purpose, it is enough to prove $\theta_i = \sqrt{1 - \alpha_i/\gamma_i} \leqslant c < 1$

with c constant and independent of n for every level $i = 0, \dots, m-1$. At the moment we consider for simplicity only $\mu = \pm 2$, but at the end we will show that we can extend the proof for every nonzero $\mu \in \mathbb{R}$. At the first level with f_0 and $p_0 = p$ we are in the hypotheses of Theorem 2.6, therefore, at the first level, the TGM converges with convergence rate $\theta_0 < 1$. Using the function relation (2.13) to find f_{i+1} from f_i and p_i, we have to distinguish between q even and odd.

- *q even:* In this case we obtain $f_1 = f_0$ so that $p_1 = p_0$ satisfies again the conditions (2.15) and then $\theta_1 = \theta_0$. Iteratively $f_{i+1} = f_i = \cdots = f_0$ with $p_i = \cdots = p_0 = p$ for $1 \leqslant i \leqslant m-1$ and $\theta_i = \theta_0$.
- *q odd:* Here we have $f_1 = 2f_0$, then $p_1 = p_0$ satisfies again the conditions (2.15) and iteratively $f_{i+1} = 2f_i = \cdots = 2^{i+1} f_0$ with $p_i = \cdots = p$ for $1 \leqslant i \leqslant m-1$. Even if f_i changes at each level, in the computation of θ_i the factor 2^i is simplified out and then $\theta_i = \theta_0$ for every i.

If $\mu \neq \pm 2$, as in the case of q odd, we obtain $f_j = \xi^j f_0$, $\xi \in \mathbb{R}$ and nonzero ξ, but again $\theta_i = \theta_0$.

In conclusion, with $c = \theta_0$ we have $\theta_i = c < 1$ for every level $i = 0, \dots, m-1$, i.e., the level independency property is satisfied. □

Remark 3.3. The previous proposition can be generalized to every function f_0 that vanishes at the origin with a zero of finite order. In particular, in this case, the level independency holds under the same TGM optimality conditions (2.15) and does not require more restrictive conditions.

Now we present an example where the projectors satisfy the previous proposition but are not sufficient for ensuring the V-cycle optimality. Moreover we will see that a slight modification of the proposed projectors will be enough for an optimal MGM convergence rate. We perform only a Richardson post-smoother iteration with $\omega = 1/\max(f_0)$ and MGM is stopped when $\|\mathbf{r}_0^{(k)}\|_2 \leq 10^{-11}\|\mathbf{b}\|_2$. From the fourth derivative discretization by finite differences and appropriate boundary conditions, we obtain a system with coefficient matrix $T_n(f_0)$, where

$$f_0(x) = (2 - 2\cos(x))^2.$$

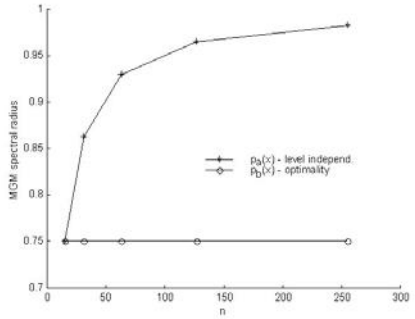
We consider its τ version $\tau_{n_0}(f_0)$ (which corresponds to the natural τ preconditioner of $T_{n_0}(f_0)$). Therefore, defining the projector at each level through the trigonometric polynomial

$$p(x) = 2 + 2\cos(x),$$

we remark that the hypotheses of Proposition 3.2 are fulfilled and then the level independency property stands. From the numerical application of the corresponding V-cycle algorithm to the system $\tau_{n_0}(f_0)\,\mathbf{x} = \mathbf{b}$ with the proposed projector, we observe that the iteration number grows (almost linearly) as the dimension n (refer to Table 3.1). Therefore, the proposed method is not optimally convergent while the level independency holds true. From the same table we can see that leaving unchanged the post-smoother and increasing the projector degree by 1, it is possible to recover the MGM optimality. The last column in Table 3.1 stresses as the fundamental choice the projector and not the smoother, indeed, also increasing the Richardson iteration number and adding some conjugate gradient (CG) iterations as post-smoother (accelerator), the MGM iteration number diverges as the problem dimension tends to infinity. We observe a similar behavior in image restoration problems: Compare these results with section 7.5, especially Table 7.8, and with [6, 18].

TABLE 3.1
MGM iteration number in the case of natural τ preconditioner for the monodimensional fourth derivative.

n	1 Richardson iteration with $\omega = 1/\max(f_0)$		$p(x) = 2 + 2\cos(x)$
	$p(x) = 2 + 2\cos(x)$ level independency holds but the MGM is not optimal	$p(x) = (2 + 2\cos(x))^2$ MGM optimality and level independency are satisfied	2 Richardson with $\omega = 1/\max(f_0)$ and 2 CG iterations
$2^7 - 1$	283	83	113
$2^8 - 1$	510	83	196
$2^9 - 1$	899	83	299
$2^{10} - 1$	1541	83	475



n	MGM_0 spectral radius	
	$p_a(x)$	$p_b(x)$
15	0.75	0.75
31	0.8629	0.75
63	0.9297	0.75
127	0.9647	0.75
255	0.9823	0.75
511	0.9912	0.75

FIG. 3.1. *Spectral radius of MGM_0 with $p(x) = p_a(x) = 2 + 2\cos(x)$ (level independency but not optimality) and $p(x) = p_b(x) = (2 + 2\cos(x))^2$ (level independency and optimality).*

The difference between level independency and MGM optimality is underlined also from Figure 3.1, where it is shown the spectral radius of the MGM iteration matrix calculated by using recurrence (2.3). In our V-cycle algorithm, we solve the system at dimension 7 by a direct method. Therefore, at dimension 15 we have that the MGM is reduced to the TGM and we notice that the MGM with $p(x) = p_a(x) = 2+2\cos(x)$ has the same spectral radius as the MGM with $p(x) = p_b(x) = (2+2\cos(x))^2$, due to the optimality of TGM.

It is starting from these remarks that in the next section we propose an optimal MGM and we prove its optimal behavior under mild assumptions on the symbol f_0.

4. Proof of convergence and optimality: The scalar case. We now show a way for satisfying the assumptions of Theorem 2.3, in their strong version (2.11a) and (2.11b). The first inequality (2.11a) is quite simple (i.e., polynomial) so it can be handled as in [27, 31]. The second is more difficult to show and it represents one of the main contributions of the paper.

4.1. How to fulfill the smoothing property. We start with a result which is a slight variation of analogous propositions in [27, 31].

PROPOSITION 4.1. *For every $i = 0, \ldots, m-1$, let $A_i = \mathcal{A}_{n_i}(f_i)$ with $\mathcal{A} \in \{\mathcal{C}, \mathcal{H}, \tau\}$ as in (1.4), f_i being nonnegative, and let ω_i be such that $0 < \omega_i < 2/\|f_i\|_\infty$. If we choose α_i fulfilling $\alpha_i \leqslant \omega_i(2 - \omega_i\|f_i\|_\infty)$ and if we define $R_i = I_{n_i} - \omega_i A_i$, then*

$$\|R_i\,\mathbf{x}\|_{A_i}^2 \leqslant \|\mathbf{x}\|_{A_i}^2 - \alpha_i\,\|\mathbf{x}\|_{A_i^2}^2 \tag{4.1}$$

holds true for every $\mathbf{x} \in \mathbb{C}^n$.

Proof. The essential steps for proving (4.1) can be found in [27, 31]. We just observe that the best bound to α_i is $1/\|f_i\|_\infty$ and it is obtained by taking $\omega_i = \omega_i^* = 1/\|f_i\|_\infty$. ∎

4.2. How to fulfill the approximation property. We still have to show how to satisfy the more intricate MGM hypothesis (2.11b). We will consider the following set of conditions,

$$p_i^2(x) + p_i^2(\pi + x) > 0, \qquad \limsup_{x \to x_0} \left| \frac{p_i(\pi + x)}{f_i(x)} \right| < +\infty, \qquad i = 0, \ldots, m-1, \tag{4.2}$$

to hold for every x and where x_0 is the unique zero of f_i of order $2q$. We observe that the conditions above are stronger than those (2.15) considered for the TGM method: a qualitative reasoning behind it is contained in section 3 and concerns the fact that the level independency does not imply the MGM optimality.

It follows that p_i must possess a unique zero of the same (or higher) order as f. We will choose p_i as follows:

$$p_{x_0,q}^{\mathcal{C},\mathcal{H}}(x) = \big[1 + \cos(x - x_0)\big]^q, \qquad p_{x_0,q}^{\tau}(x) = \big[\cos(x_0) + \cos(x)\big]^q \ \text{ if } x_0 \in \{0, \pi\}. \tag{4.3}$$

In addition, if f has a zero at $x_0 \notin \{0, \pi\}$ and we are in the τ case, then f is even and also has a zero at $2\pi - x_0$: in that case of two zeros we choose $p_{\pm x_0,q}^{\tau}(x) = \big[\cos(x_0) + \cos(x)\big]^{2q}$. Finally, we consider a product of some of these basic polynomials in the general multiple-zeros case (see also [15, 27, 31]).

We will also use the following factorization result.

PROPOSITION 4.2. *Let f be a trigonometric polynomial such that $f(x_0) = 0$ and $f(x) > 0$ whenever $x \not\equiv x_0 \mod 2\pi$. Then there exists a positive trigonometric polynomial ψ such that*

$$f(x) = [1 - \cos(x - x_0)]^q \cdot \psi(x) \tag{4.4}$$

and $2q$ is the order of f at x_0.

In the rest of the subsection and in section 4.3 we will focus our attention on the important case where the symbol has a unique zero at $x = 0$ (this includes various discretized boundary values problems) with the exception of Proposition 4.5 and Remark 4.7: the more general case of a zero not at $x = 0$ will be briefly treated in sections 4.4 and 4.5.

Proposition 4.2 ensures a suitable factorization for our generating function f_0, i.e., $f_0(x) = [1 - \cos(x)]^q \psi_0(x)$, ψ_0 being a positive trigonometric polynomial, when dealing with τ matrices. In the case of the circulant and Hartley algebras, we must consider the one rank correction displayed in (1.5) in order to force the invertibility. Therefore, by exploiting relation (4.4), we have $f_0(x) + c_0 \chi_{2\pi\mathbb{Z}}(x) = [1 - \cos(x)]^q \psi_0(x) + c_0 \chi_{2\pi\mathbb{Z}}(x)$. In order to get a uniform lower bound to α_i/β_i (in particular to find an upper bound for the left side of (4.8)), it seems convenient to obtain such a factorization for every generating function f_i. We find the desired result by using Proposition 2.5.

PROPOSITION 4.3. *Under the same assumptions of Proposition 2.5, let q be a positive integer and let us suppose $f_0(x) = [1 - \cos(x)]^q \psi_0(x) + c_0 \chi_{2\pi\mathbb{Z}}(x)$, with ψ_0 being a positive trigonometric polynomial and with $c_0 = f_0(w_1^{[n_0]})$ in the circulant and Hartley cases and with $c_0 = 0$ in the τ case; define also $p_i(x) = \sqrt{2}\,[1 + \cos(x)]^q +$*

$d_i\,\chi_{2\pi\mathbb{Z}}(x) = p(x) + d_i\,\chi_{2\pi\mathbb{Z}}(x)$ for each $i = 0, \dots, m-1$. Then each generating function f_i satisfies $f_i(x) = \tilde{f}_i(x) + c_i\chi_{2\pi\mathbb{Z}}(x)$, $\tilde{f}_i(x) = [1-\cos(x)]^q\psi_i(x)$ with the sequences $\{\psi_i\}$ and $\{c_i\}$ defined as

$$\begin{cases} \psi_{i+1} = \Phi_q(\psi_i), \\ c_{i+1} = \frac{1}{2}c_i p_i^2(0), \end{cases} \qquad i = 0, \dots, m-1,$$

where Φ_q is an operator such that

$$\big[\Phi_q(\psi)\big](x) = \frac{1}{2^{q+\frac{1}{2}}}\Big[(p\,\psi)\Big(\frac{x}{2}\Big) + (p\,\psi)\Big(\pi + \frac{x}{2}\Big)\Big]. \tag{4.5}$$

Moreover, each $\tilde{f}_i$ is a trigonometric polynomial that vanishes only at $2\pi\mathbb{Z}$ with the same order $2q$ as f_0.

Proof. Taking into account the expression of $p(x) = \sqrt{2}\,[1+\cos(x)]^q$, the result is a direct consequence of Proposition 2.5 and relation (2.14). □

PROPOSITION 4.4. *Under the same assumptions of Proposition* 2.5*, let q be a positive integer and let us suppose $f_0(x) = [1-\cos(x)]^q\psi_0(x) + c_0\chi_{2\pi\mathbb{Z}}(x)$, with ψ_0 being a positive trigonometric polynomial and with $c_0 = f(w_1^{[n_0]})$ in the circulant and Hartley cases and with $c_0 = 0$ in the τ case; also define $p_i(x) = \sqrt{2}\,[1+\cos(x)]^q + d_i\,\chi_{2\pi\mathbb{Z}}(x) = p(x) + d_i\,\chi_{2\pi\mathbb{Z}}(x)$ for each $i = 0, \dots, m-1$. Then we can choose numbers d_i such that, setting $\tilde{f}_i(x) = [1-\cos(x)]^q\psi_i(x)$, we have $f_i(x) = \tilde{f}_i(x) + c_i\chi_{2\pi\mathbb{Z}}(x)$ with $c_i = 0$ in the τ case and with $c_i = f_i(w_0^{[n_i]}) = \tilde{f}_i(w_1^{[n_i]}) > 0$ in the case of circulants and Hartley matrices.*

Proof. In the τ setting we can choose $d_i = 0$. Therefore, since $c_0 = 0$ there is nothing to prove. In the remaining cases, the result follows from the relations $f_i(0) = \tilde{f}_i(0) + c_i = c_i$, $c_{i+1} = \frac{1}{2}c_i p_i^2(0) = \frac{1}{2}c_i(\sqrt{2}\,2^q + d_i)^2$ and from the fact that $c_0 = f(w_1^{[n_0]})$: more specifically we have

$$d_i = \sqrt{\frac{2f_{i+1}\left(\frac{2\pi}{n_{i+1}}\right)}{f_i\left(\frac{2\pi}{n_i}\right)}} - \sqrt{2}\,2^q. \qquad \square$$

Propositions 4.3 and 4.4 will allow us to find bounds for the constants α_i and β_i involved in (2.11).

We now have the tools for defining a really recursive V-cycle technique (as explained in Proposition 4.5) and for proving that we can satisfy the approximation property (Proposition 4.6).

PROPOSITION 4.5. *Let $A_i = \mathcal{A}_{n_i}(f_i)$, $P_{i+1}^i = K_{i+1}^i\mathcal{A}_{n_i}(p_i)$, with f_i being a nonnegative polynomial (also even in the τ case) and p_i satisfying conditions* (4.2) *(also even in the τ case).*

1. *The projected matrix A_{i+1} coincides with $\mathcal{A}_{n_{i+1}}(f_{i+1})$, where f_{i+1} has the expression reported in* (2.13)*.*
2. *If $x_0 \in [-\pi, \pi]$ is a zero of $f_i(x)$ then f_{i+1} has a corresponding zero $y_0 = 2x_0$.*
3. *f_i and f_{i+1} have the same number of zeros, i.e., for any zero $y_0 \in [-\pi, \pi]$ of f_{i+1} there exists a unique zero of f_i such that the relations in the preceding item holds true.*
4. *The order of the zero y_0 of f_{i+1} is exactly the same as the one of the zero x_0 of f_i so that at the lower level the new projector is easily defined in the same way.*

PROPOSITION 4.6. *For every $i = 0, \dots, m-1$, let $A_i = \mathcal{A}_{n_i}(f_i)$ with $\mathcal{A} \in \{\mathcal{C}, \mathcal{H}, \tau\}$ as in (1.4) and let f_i be as in Proposition 4.3. Let $P^i_{i+1} = K^i_{i+1} \cdot \mathcal{A}_{n_i}(p_i)$ and let us define CGC_i as in (2.5). Assume that $p_i(x) = \tilde{p}_i(x) + d_i\, \chi_{2\pi\mathbb{Z}}(x)$ with $\tilde{p}_i$ fulfilling (4.2) (also even in the τ case) and with d_i as in Proposition 4.4 (for instance, take $\tilde{p}_i(x) = \sqrt{2}\,[1+\cos(x)]^q$ as in Proposition 4.3). Then for every $i = 0, \dots, m-1$, there exists a real and positive value β_i such that*

$$\|CGC_i\, \mathbf{x}\|^2_{A_i} \leqslant \beta_i\, \|\mathbf{x}\|^2_{A_i^2}, \qquad \mathbf{x} \in \mathbb{C}^{n_i}. \tag{4.6}$$

Proof. Relation (4.6) can be rewritten in matrix form as

$$CGC_i^H A_i\, CGC_i \leqslant \beta_i A_i^2.$$

By straightforward calculation we have $CGC_i^H A_i\, CGC_i = A_i\, CGC_i$, and hence (4.6) holds if and only if $A_i CGC_i \leqslant \beta_i A_i^2$ is satisfied. By multiplying from both the sides by $A_i^{-1/2}$ we get

$$I_{n_i} - A_i^{1/2} (P^i_{i+1})^H \left[P^i_{i+1} A_i (P^i_{i+1})^H\right]^{-1} P^i_{i+1} A_i^{1/2} \leqslant \beta_i A_i,$$

and then, by defining $\hat{P}^i_{i+1} = P^i_{i+1} \cdot A_i^{1/2}$, we infer

$$I_{n_i} - (\hat{P}^i_{i+1})^H \left[\hat{P}^i_{i+1} (\hat{P}^i_{i+1})^H\right]^{-1} \hat{P}^i_{i+1} \leqslant \beta_i A_i, \tag{4.7}$$

where $\hat{P}^i_{i+1} = K^i_{i+1} \cdot \mathcal{A}_{n_i}\big(\hat{p}_i(x)\big)$ with $\hat{p}_i(x) = p_i(x) \cdot f_i^{1/2}(x)$. We notice that (4.7) can be found in [31] while showing the TGM approximation property for the circulant algebra, and is also contained in the proof of Lemma 3.2 in [27], while showing the same property in the τ algebra (the Hartley case is totally analogous to circulants). Thus, by performing a block diagonalization of all the involved matrices (see Lemma 3.2 in [27]), to have (4.6), it is enough to prove

$$\frac{1}{\hat{p}_i^2(x) + \hat{p}_i^2(x+\pi)} \begin{bmatrix} \hat{p}_i^2(\pi+x) & -\hat{p}_i(x)\hat{p}_i(\pi+x) \\ -\hat{p}_i(x)\hat{p}_i(\pi+x) & \hat{p}_i^2(x) \end{bmatrix} \leqslant \beta_i \begin{bmatrix} f_i(x) & \\ & f_i(\pi+x) \end{bmatrix}$$

for every $x \in \bigcup_{j \in \mathcal{I}_{n_{i+1}}} \left\{\frac{1}{2} w_j^{[n_{i+1}]}\right\}$, and once again, by following the proof of Lemma 3.2 in [27], we deduce that (4.6) is guaranteed if

$$\frac{1}{\hat{p}_i^2(x) + \hat{p}_i^2(\pi+x)} \cdot \left(\frac{\hat{p}_i^2(x)}{f_i(\pi+x)} + \frac{\hat{p}_i^2(\pi+x)}{f_i(x)}\right) \leqslant \beta_i \qquad \forall x \in \bigcup_{j \in \mathcal{I}_{n_{i+1}}} \left\{\frac{w_j^{[n_{i+1}]}}{2}\right\}.$$

Therefore, in terms of the involved generating functions, we obtain that the following conditions have to be satisfied:

$$\frac{1}{\dfrac{p_i^2(x)}{f_i(\pi+x)} + \dfrac{p_i^2(\pi+x)}{f_i(x)}} \cdot \left(\frac{p_i^2(x)}{f_i^2(\pi+x)} + \frac{p_i^2(\pi+x)}{f_i^2(x)}\right) \leqslant \beta_i \quad \forall x \in \bigcup_{j \in \mathcal{I}_{n_{i+1}}} \left\{\frac{w_j^{[n_{i+1}]}}{2}\right\}. \tag{4.8}$$

Finally, we observe that the first inequality in (4.2) implies the uniform boundedness (with respect to n_i and to x) of the term

$$\frac{1}{\dfrac{p_i^2(x)}{f_i(\pi+x)}+\dfrac{p_i^2(\pi+x)}{f_i(x)}},$$

while the second inequality in (4.2) implies the uniform boundedness (with respect to n_i and to x) of the term

$$\left(\frac{p_i^2(x)}{f_i^2(\pi+x)}+\frac{p_i^2(\pi+x)}{f_i^2(x)}\right),$$

and therefore the proof is over with β_i being the products of the two constants realizing the above mentioned bounds. □

Remark 4.7. The statement in Proposition 4.6, namely relation (4.6), holds unchanged in the more general setting where the zero x_0 is not 0. It is sufficient to show that A_i is nonsingular and indeed the rest of the proof of Proposition 4.6 will remain the same. Let $A_i = \mathcal{A}_{n_i}(f_i)$ with $\mathcal{A} \in \{\mathcal{C}, \mathcal{H}, \tau\}$ for $i = 0, \ldots, m$ and let x_0 be the unique zero of f_0 in $[0, 2\pi)$ (in the τ case f_0 is even and has also a zero at $2\pi - x_0$).

1. If $x_0 \notin \bigcup_{j\in\mathcal{I}_{n_0}}\{w_j^{[n_0]}\}$ (also $2\pi - x_0 \notin \bigcup_{j\in\mathcal{I}_{n_0}}\{w_j^{[n_0]}\}$ in the τ case), then, by Proposition 4.5, f_i vanishes only at $x_i \notin \bigcup_{j\in\mathcal{I}_{n_i}}\{w_j^{[n_i]}\}$ (also at $2\pi - x_i \notin \bigcup_{j\in\mathcal{I}_{n_i}}\{w_j^{[n_i]}\}$ in the τ case) and therefore A_i is nonsingular for every $i = 0, \ldots, m$.
2. If $\exists\, j \in \mathcal{I}_{n_0} : x_0 = w_j^{[n_0]}$ (also $2\pi - x_0 = w_{n+1-j}^{[n_0]}$ in the τ case), we proceed as in the case $x_0 = 0$ (see Propositions 4.3 and 4.4). We fix $\tilde{f}_i(x) = (1-\cos(x - x_i))^q$, $c_0 = \min\{f_0(w_{j-1}^{[n_0]}), f_0(w_{j+1}^{[n_0]})\}$ and $f_i(x) = \tilde{f}_i(x) + c_i\,\chi_{x_0+2\pi\mathbb{Z}}(x)$ (also $f_i(x) = \tilde{f}_i(x) + c_i\,\chi_{-x_0+2\pi\mathbb{Z}}(x)$ in the τ case), then, by Proposition 4.5, $\tilde{f}_i$ vanishes at $x_i = w_j^{[n_i]}$ (also at $w_{n+1-j}^{[n_0]}$ in the τ case) for $i = 0, \ldots, m$. The quantities c_i, d_i and $p_i(x)$ are calculated as in Propositions 4.3 and 4.4, where 0 is replaced by x_0. In this case A_i is again nonsingular for every $i = 0, \ldots, m$.

4.3. MGM optimal convergence (i.e., verification of the inf–min condition). In Propositions 4.1 and 4.6 we have proven that for every i (independent of $n = n_0$) the constants α_i and β_i are absolute values not depending on $n = n_0$ but only depending on the functions f_i and p_i. However, in order to fulfill conditions (2.11a) and (2.11b) with δ independent of n (which in turn imply the MGM optimal convergence by Theorem 2.3), we should prove the following inf–min condition:

$$\delta = \inf_n \min_{1\leqslant m\leqslant\phi(n)} \min_{0\leqslant i\leqslant m} \frac{\alpha_i}{\beta_i} = \inf_n \min_{0\leqslant i\leqslant\phi(n)} \frac{\alpha_i}{\beta_i} > 0. \tag{4.9}$$

Here $\phi(n)$ is the maximal number of possible recursion levels and it equals $\log_2(n)$ for circulants and Hartley matrices and coincides with $\log_2(n+1)$ for τ matrices. In the following we will consider the case where the trigonometric polynomial f_0 is positive in the interval $(0, 2\pi)$ and takes the zero value at the origin, and we will demonstrate the inf–min condition (4.9).

In the following, for a given function f, we will write $M_f = \sup_x |f|$, $m_f = \inf_x |f|$ and $\mu_\infty(f) = M_f/m_f$. In (2.11a) we simply find $\alpha_i\big(\omega_i^* = \|f_i\|_\infty^{-1}\big) = \|f_i\|_\infty^{-1} \geqslant$

$1/(2^q M_{\psi_i})$, while (from $p(x) = \sqrt{2}[1+\cos(x)]^q$ it follows that the range of $p(x) + p(\pi+x)$ is $[\sqrt{2}\cdot 2, \sqrt{2}\cdot 2^q]$) in order to get an upper bound for the left-hand side in (4.8), if $x \in (0, 2\pi)$ we obtain

$$\frac{\dfrac{p_i^2(x)}{f_i^2(\pi+x)} + \dfrac{p_i^2(\pi+x)}{f_i^2(x)}}{\dfrac{p_i^2(x)}{f_i(\pi+x)} + \dfrac{p_i^2(\pi+x)}{f_i(x)}} = \sqrt{2}\frac{\dfrac{1}{\psi_i^2(\pi+x)} + \dfrac{1}{\psi_i^2(x)}}{\dfrac{p_i(x)}{\psi_i(\pi+x)} + \dfrac{p_i(\pi+x)}{\psi_i(x)}} \leqslant \sqrt{2}\frac{\dfrac{2}{m_{\psi_i}^2}}{\dfrac{p(x)+p(\pi+x)}{M_{\psi_i}}} \leqslant \frac{M_{\psi_i}}{m_{\psi_i}^2}$$

so $\beta_i = M_{\psi_i}/m_{\psi_i}^2$ works fine, while if $x = 0$ we have also to require $1/f_i(\pi) \leqslant \beta_i$ to ensure that inequality (2.11b) is satisfied: more precisely, at $x = 0$ (by (4.8), the case is of interest only for circulants and Hartley matrices since $x = 0$ is not a grid point for τ matrices), we have

$$\begin{aligned} p_i(0) &= \sqrt{2}\,2^q + d_i, \\ f_i(0) &= c_i > 0, \\ p_i(\pi) &= 0, \\ f_i(\pi) &> 0, \end{aligned}$$

and therefore (4.8) holds at $x = 0$ with any constant β_i such that $1/f_i(\pi) \leqslant \beta_i$. Since $(M_{\psi_i}/m_{\psi_i}^2)\cdot f_i(\pi) \geqslant f_i(\pi)/m_{\psi_i} \geqslant 1$, it follows that $\beta_i^* = M_{\psi_i}/m_{\psi_i}^2$ is the best value. As a consequence, it follows that

$$\frac{\alpha_i}{\beta_i} \geqslant \frac{1}{2^q M_{\psi_i}} \cdot \frac{m_{\psi_i}^2}{M_{\psi_i}} = \frac{1}{2^q \mu_\infty^2(\psi_i)}. \tag{4.10}$$

Therefore, to enforce the inf–min condition (4.9), it is enough to prove the existence of an absolute constant L such that $\mu_\infty(\psi_i) \leqslant L < +\infty$ uniformly to deduce that $\|MGM_0\|_{A_0} \leqslant \sqrt{1-2^{-q}L^{-2}} < 1$: the latter follows from the next proposition.

PROPOSITION 4.8. *Under the same assumptions of Proposition* 4.3*, let ψ_0 be a positive polynomial and let us define $\psi_i = [\Phi_q]^i(\psi)$ for every $i \in \mathbb{N}$, where Φ_q is the linear operator defined as in* (4.5)*. Then there exists a positive polynomial $\psi_\infty \in \mathbb{R}_{q-1}$ such that ψ_i uniformly converges to ψ_∞, and moreover there exists a positive real number L such that $\mu_\infty(\psi_i) \leqslant L$ for every $i \in \mathbb{N}$.*

Proof. The proof is organized into two parts.

Part A. From the definition of the operator Φ_q in (4.5) and from the assumptions on the polynomials p_i (see Proposition 4.3), it follows that the positivity (and the boundedness) of ψ_0 implies the positivity (and the boundedness) of ψ_i for every $i \in \mathbb{N}$, i.e., there exist positive constants L_i such that

$$\mu_\infty(\psi_i) \leqslant L_i. \tag{4.11}$$

Part B. We give a linear algebra proof of the fact that, starting a polynomial ψ_0 such that $\psi_0(0) > 0$, the operator Φ_q in (4.5) has a strictly positive fixed point belonging to $\mathbb{R}_q$, and therefore there exists a constant L_∞ such that

$$\lim_{i\to\infty} \mu_\infty(\psi_i) = L_\infty. \tag{4.12}$$

Therefore the second result ($\mu_\infty(\psi_i) \leqslant L$ for $i \in \mathbb{N}$ and for a pure constant $L > 0$) will be a straightforward consequence of (4.11) and of (4.12) which, in turn, is a

consequence of the uniform convergence of the sequence ψ_i and of the fact that its limit is a strictly positive function. The latter is what we are going to prove.

From (2.14) it follows that $f_i \in \mathbb{R}_{T_i}$ with $T_{i+1} = q + \lfloor \frac{T_i}{2} \rfloor$ if $f_0 \in \mathbb{R}_{T_0}$, and hence, since $f_i = \psi_i[1-\cos(x)]^q$, there exists an index $j \in \mathbb{N}$ such that $\psi_i \in \mathbb{R}_q$ when $i \geqslant j$ and we can suppose $\psi_i \in \mathbb{R}_q$. We demonstrate a bit stronger result, i.e.,

$$\begin{cases} \psi \in \mathbb{R}_q \\ \psi(0) > 0 \end{cases} \qquad \Rightarrow \qquad \exists \psi^* \in \mathbb{R}_q : [\Phi_q]^i(\psi) \xrightarrow{\text{uniformly}} \psi^*.$$

As Φ_q is linear, by expressing the problem in the basis $\{e^{-\mathbf{i}qx}; \dots ; e^{\mathbf{i}qx}\}$ and by denoting by $\bar{\Phi}_q$ the matrix representing Φ_q in such a basis, the preceding implication is equivalent to proving that

$$\begin{cases} \mathbf{a} \in \mathbb{C}^{2q+1} \\ \sum_{j=1}^{2q+1} a_j > 0 \end{cases} \qquad \Rightarrow \qquad \exists \mathbf{a}^* \in \mathbb{C}^{2q+1} : \left[\bar{\Phi}_q\right]^i \mathbf{a} \longrightarrow \mathbf{a}^*. \tag{4.13}$$

If (4.13) holds true, then there exists $\psi^* \in \mathbb{R}_q$ (defined by $\psi^*(x) = \sum_{|j| \leqslant q} a^*_{j+q+1} e^{\mathbf{i}jx}$) such that $\psi_i \longrightarrow \psi^*$ uniformly and $\Phi_q(\psi^*) = \psi^*$. Moreover, from the assumptions on p_i (see Proposition 4.3), we have $p_i(\pi) = 0$, $p(0) = 2^{q+\frac{1}{2}}$ and therefore, by (4.5), we have

$$\psi_{i+1}(0) = \Phi_q(\psi_i)(0) = \frac{p(0)}{2^{q+\frac{1}{2}}}\psi_i(0) = \psi_i(0).$$

Thus $\psi^*(0) = \psi_0(0) > 0$. The last condition ensures $\psi^* > 0$, because, from $\psi^*(\bar{x}) = 0$ and from the definition of $\Phi_q(\cdot)$, it follows $\psi^*(\bar{x}/2^s) = 0$ for every $s \in \mathbb{N}$ (use (4.5)), and this is clearly impossible because ψ^* is continuous and therefore

$$\lim_{s \to \infty} \psi^*(\bar{x}/2^s) = \psi^*(0) > 0.$$

We still have to show (4.13). In actuality, (4.13) follows if we demonstrate that $\bar{\Phi}_q$ has one eigenvalue equal to 1 with algebraic multiplicity 1 and positive eigenvector $\mathbf{a}^*$, while all the other eigenvalues λ_i enjoy the relation $|\lambda_i| < 1$: to this aim we will use the Perron–Frobenius theorem [19, 36]. Let us look at $\bar{\Phi}_q$. We define $b_j^{(q)}$, $|j| \leqslant q$, as the Fourier coefficient of $\frac{1}{2^{q+\frac{1}{2}}}p(x)$ (i.e., of $\cos^{2q}(x/2)$):

$$b_j^{(q)} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{1}{2^{q+\frac{1}{2}}} p(x) e^{\mathbf{i}jx}\, dx = \frac{(2q)!}{4^q(q-j)!(q+j)!} > 0, \qquad b_j^{(q)} = b_{-j}^{(q)}.$$

It holds that $p(x)e^{\mathbf{i}kx} = \sum_{j=k-q}^{k+q} b_{j-k}^{(q)} e^{\mathbf{i}jx}$ and hence $\bar{\Phi}_q$ has (by (2.14)) the following matrix form:

(4.14)

$$\Phi_q\left(e^{\mathbf{i}kx}\right) = 2\sum_{j=\lceil \frac{k-q}{2} \rceil}^{\lfloor \frac{k+q}{2} \rfloor} b_{2j-k}^{(q)} e^{\mathbf{i}jx} \quad \Rightarrow \quad \overline{\Phi}_q = 2 \begin{bmatrix} b_{-q}^{(q)} & & & & \\ \vdots & \boxed{\begin{matrix} b_{1-q}^{(q)} & b_{-q}^{(q)} & & \\ \vdots & \ddots & \ddots & \\ b_{q-1}^{(q)} & & & b_{1-q}^{(q)} \\ & \ddots & \ddots & \vdots \\ & & & b_q^{(q)} \quad b_{q-1}^{(q)} \end{matrix}} & b_{-q}^{(q)} \\ \vdots & & \vdots \\ b_q^{(q)} & & b_q^{(q)} \end{bmatrix}_{-q:q \times -q:q}.$$

Nonvanishing entries in the kth column are the coefficients $2b_j^{(q)}$ such that $j \equiv k \pmod 2$. We observe $b_q^{(q)} = b_{-q}^{(q)} = 4^{-q} < 1$ and then we have only to check the behavior of the eigenvalues of the submatrix M with both indices ranging from $-q+1$ to $q-1$ (this matrix is the one displayed in the inner box of (4.14)). The corresponding analysis is now straightforward: the vector of all ones is an eigenvector of M^T related to the eigenvalue 1 (because of the left side in (4.14)); $\|M\|_\infty = 1$, $M_{ij} \geqslant 0$ and M is irreducible. Finally, the result follows from the Perron–Frobenius theorem applied to the matrix M. ∎

We remark that the previous result on the limit polynomial ψ^* can be refined a little bit. Indeed, it belongs to $\mathbb{R}_{q-1}$ (instead of $\mathbb{R}_q$) since the eigenvector $\mathbf{a}^*$ of $\overline{\Phi}_q$ related to the dominating eigenvalue $\lambda = 1$ is of the form

$$\begin{pmatrix} 0 \\ \hat{\mathbf{a}} \\ 0 \end{pmatrix},$$

where $\hat{\mathbf{a}}$ is the positive eigenvector of M associated with the dominating eigenvalue $\lambda = 1$.

THEOREM 4.9. *Let f be a trigonometric polynomial, positive in $(0, 2\pi)$ and vanishing at 0 with order $2q$ (also even in the τ case); let us fix integers k_0, m such that $0 < m < k_0$, and let us define n_i, $i = 0, \ldots, m$, as in (2.12).*

For every $i = 0, \ldots, m-1$, define also the following quantities: $p_i(x) = \sqrt{2}\big[1 + \cos(x)\big]^q + d_i\, \chi_{2\pi\mathbb{Z}}(x)$ with d_i as Proposition 4.4, K_{i+1}^i as in (2.12), $P_{i+1}^i = K_{i+1}^i \mathcal{A}_{n_i}(p_i)$ with $\mathcal{A} \in \{\mathcal{C}, \mathcal{H}, \tau\}$ as in (1.4), and $\mathcal{R}_i$ as in (2.2) with $R_i = I_{n_i} - A_{n_i}/\|f_i\|_\infty$.

If we set $A_0 = \mathcal{A}_{n_0}(f + c_0 \chi_{2\pi\mathbb{Z}})$ with $c_0 = f(w_1^{[n_0]})$ in the circulant and Hartley cases and with $c_0 = 0$ in the τ case, and we consider $\mathbf{b} \in \mathbb{C}^{n_0}$, then the V-cycle algorithm defined in (2.1) converges to the solution of $A_0\mathbf{x} = \mathbf{b}$ and is optimal (in the sense of Definition 1.1).

Proof. From Proposition 4.2 we know that

$$f(x) = [1 - \cos(x)]^q \cdot \psi(x)$$

for some positive polynomial ψ. Now, it is enough to observe that the MGM optimal convergence stated in Theorem 2.3 is implied by the inf–min condition (4.9) which, in turn, by (4.10), is implied by the uniform boundedness of the quantities $\mu_\infty(\psi_i)$ and the latter has been proven in Proposition 4.8. ∎

4.4. The case of a unique zero at $x_0 \neq 0$: Circulant and Hartley algebras. We now consider matrices belonging to the circulant and Hartley algebras, whose generating function f_0 vanishes in a generic point x_0. We remark that Proposition 4.2 ensures $f_0 = [1 - \cos(x - x_0)]^q \psi_0(x - x_0)$. Consequently, as in the previous situation ($x_0 = 0$), we obtain a similar result.

PROPOSITION 4.10. *Under the same assumptions of Proposition 2.5, let $f_0(x) = [1 - \cos(x - x_0)]^q \psi_0(x - x_0)$ with q positive and integer and let ψ_0 be a positive trigonometric polynomial. By defining $x_{i+1} = 2x_i \pmod{2\pi}$ and $p_i(x) = \sqrt{2}\,[1 + \cos(x - x_i)]^q$ for every $i = 0, \ldots, m-1$, we deduce that the generating functions f_i enjoy the following relation:*

$$f_i(x) = [1 - \cos(x - x_i)]^q \psi_i(x - x_i).$$

Proof. It suffices to write $f_{i+1}(x) = \frac{1}{2}[(p_i^2 f_i)(\frac{x - 2x_i}{2}) + (p_i^2 f_i)(\pi + \frac{x - 2x_i}{2})]$ and to apply the statement contained in Proposition 4.3. ∎

In such a situation the functions f_i will not converge, but the values $\mu_\infty(\psi_i)$ remain unchanged and the latter is enough to prove the MGM optimal convergence (see (4.10)).

4.5. The case of a unique zero at $x_0 \neq 0$: the τ algebra. Since the generating function f_0 must be also even, it follows that the unicity of the zero $x_0 \neq 0$ implies that f_0 has to vanish at $x_0 = \pi$. In addition, it is worth mentioning that when the coefficient matrix is $A = \tau_{n_0}(f_0)$ and $f_0(\pi) = 0$ and is positive elsewhere in $(0, 2\pi)$, from relation (2.13), it follows that the function f_1 has a unique zero at 0. Since the MGM optimality, for functions having a unique zero at the origin, has been proven (see Theorem 4.9), it easily follows that the MGM optimal convergence stands for the case of a unique zero at π. Indeed, looking at the MGM applied to $\tau_{n_1}(f_1)$, it holds that (2.9) is satisfied with

$$\bar{\delta} \ \text{ independent of } \ n_1 = \frac{n_0 - 1}{2} \qquad \forall i = 1, \dots m-1.$$

Therefore, Theorem 2.3 holds with $\delta = \min\{\delta_0, \bar{\delta}\}$, which is constant and independent of n_0, i.e., the MGM is optimal.

We point out that the case of generating function that vanishes at π with respect to each variable is particularly important in applications. In fact, certain integral equations when discretized lead to matrices belonging to this class. For instance, the signal restoration leads to the case of $f(\pi) = 0$, while for the super-resolution problem and image restoration we have $f(\pi, \pi) = 0$ [8]. Therefore, it is interesting to stress that the application of the V-cycle algorithm is such that a discretized integral problem is projected, at the lower level, into another which is spectrally and structurally equivalent to a discretized differential problem.

Finally, we observe that the case of two zeros x_0 and $2\pi - x_0$ for $x_0 \notin \{0, \pi\}$ is not different from the case of a unique zero since Proposition 4.2 holds with $[\cos(x_0) - \cos(x)]^{2q}$ in place of $[1 - \cos(x - x_0)]^q$ and (4.10) is satisfied as well.

5. The multilevel case. We briefly describe our choice of projectors and smoothers in the multilevel case and we indicate how to generalize the proof of MGM optimal convergence (for the TGM the optimality has been already proven in [27, 31]).

The smoothing iteration is formally defined as in the unilevel case. The projectors are constructed as $U_{i+1}^i \mathcal{A}_{n_i}(p_i)$, where $n_i = ((n_i)_1, \dots, (n_i)_d)$, the polynomial p_i is d variate polynomial and the matrix U_{i+1}^i is defined as $K_{i+1}^{i,1} \otimes \cdots \otimes K_{i+1}^{i,d}$ with $K_{i+1}^{i,j}$ being the $(n_{i+i})_j \times (n_i)_j$ unilevel cutting matrix related to $\mathcal{A}$ explicitly given in (2.12). If the coefficient matrix is $\mathcal{A}_{n_i}(f_i)$ with f_i having a unique zero at x_0 of order $2q$, the matrix $\mathcal{A}_{n_i}(p_i)$ is chosen with p_i such that

$$\limsup_{x \to x_0} \left| \frac{p_i(\hat{x})}{f_i(x)} \right| < +\infty, \qquad \hat{x} \in M(x), \ i = 0, \dots, m-1, \tag{5.1}$$

where

$$0 < \sum_{\hat{x} \in M(x) \cup \{x\}} p_i^2(\hat{x}), \qquad i = 0, \dots, m-1, \tag{5.2}$$

with $M(x)$ being the set of the "mirror points" of x introduced for $d = 2$ in [16]. A formal definition is the following: $\hat{x} \in M(x)$ if and only if $\hat{x} \neq x$ and $\forall j = 1, \dots, d$ it holds $\hat{x}_j \in \{(x)_j, \pi + (x)_j\}$. For $d = 1$, it is evident that the unique mirror point

is $\pi + x$, while in the general case the cardinality of $M(x)$ is $2^d - 1$. Notice that $\forall \hat{x} \in M(x)$ we have $M(\hat{x}) = \{M(x) \backslash \{\hat{x}\}\} \cup \{x\}$.

If f_i has more than one zero in $[0, 2\pi]^d$ then the corresponding polynomial p_i will be the product of the basic polynomials satisfying (5.1) and (5.2) for any single zero.

Remark 5.1. In the case of more than one zero, relation (5.2) imposes some restrictions on the zeros of f_0. First, the zeros of f_0 should be of finite order (by (5.1)) and this is true in the case of a unique zero, too. Second, if x^0 is a zero of f_0 then $f(\hat{x}) > 0$ for any $\hat{x} \in M(x^0)$; otherwise relationship (5.2) cannot be satisfied with any polynomial p_0. As in the unidimensional case the second restriction can be removed by changing the "form" of the projection that is its smaller dimension.

PROPOSITION 5.2. *Let $A_i = \mathcal{A}_{n_i}(f_i)$, $P_{i+1}^i = U_{i+1}^i \mathcal{A}_{n_i}(p_i)$, with f_i being a nonnegative polynomial (also even in the τ case) and p_i satisfying conditions* (5.1) *and* (5.2) *(also even in the τ case).*

1. *The projected matrix A_{i+1} coincides with $\mathcal{A}_{n_{i+1}}(f_{i+1})$, where*
$$2^d \hat{f}_{i+1}(x) = \sum_{\hat{x} \in M(x/2) \cup \{x/2\}} f_i(\hat{x}) p_i^2(\hat{x})$$
for $x = (x_1, \ldots, x_d) \in [-\pi, \pi]^d$.
2. *If $x^0 \in [-\pi, \pi]^d$ is a zero of $f_i(x)$, then f_{i+1} has a corresponding zero $y^0 \in [-\pi, \pi]^d$ where $y_j^0 = 2x_j^0$ with $j = 1, \ldots, d$.*
3. *f_i and f_{i+1} have the same number of zeros, i.e., for any $y^0 \in [-\pi, \pi]^d$ zero of f_{i+1} there exists a unique zero of f_i such that the relations in the preceding item holds true.*
4. *The order of the zero y^0 of f_{i+1} is exactly the same as the one of the zero x^0 of f_i so that at the lower level the new projector is easily defined in the same way.*

The preceding proposition gives us the necessary tools for talking about the MGM optimal convergence. Indeed it is easy to verify that the proofs of Propositions 4.1 and 4.6 are directly generalized to the multilevel setting. The difficult part concerns relation (4.10), which is strongly based on the factorization result of Proposition 4.2. In actuality, we notice that relation (4.4) is inherently one-dimensional so that the complete multilevel proof could require a different tool at this point of the reasoning: in this respect, very recently, a substantial step has been made by the first two authors by considering an additive representation of the symbols (for more details see [2]).

6. MGM techniques for multilevel Toeplitz matrices. We first observe that the discretization of elliptic boundary value problems with constant coefficients and many image restoration problems lead to Toeplitz structures in which the symbol $f = f_0$ is polynomial, nonnegative with isolated zeros, and even (with respect to every direction if f_0 is multivariate). The latter property suggests that the right starting point for generalizing the V-cycle algorithm to Toeplitz structures should be the MGM for τ matrices (see also the beginning of Hackbush's book [17]).

In the following, we generalize the V-cycle techniques previously defined for the (multilevel) τ algebra to the (multilevel) Toeplitz class using the relation (1.2) which characterizes any Toeplitz matrix as its natural τ preconditioner plus a Hankel correction. In [27] the author presents three different choices of P_{i+1}^i when the coefficient matrix $A_{n_0}(f_0)$ is Toeplitz:

(A) $P_{i+1}^i = K_{i+1}^i T_{n_i}(p_i)$,

(B) $P_{i+1}^i = K_{i+1}^i \tau_{n_i}(p_i)$,

(C) $P_{i+1}^i = K_{i+1}^i [t_i] T_{n_i}(p_i), \quad i = 0, \ldots, m-1.$

Here p_i is the projection trigonometric polynomial defined via the same conditions as in the τ algebra case for every level $i = 0, \ldots, m-1$. For the TGM we have only $i = 0$ and p_0 is such that the conditions (2.15) are satisfied. On the other hand, for the multigrid algorithm (see section 4), the polynomials p_i are chosen in such a way that the stronger conditions (4.2) are satisfied. The choice (A) is the most natural, but unfortunately the lower level matrix $A_{n_{i+1}} = P_{i+1}^i A_{n_i} (P_{i+1}^i)^H$ is not Toeplitz unless the degree of p_i does not exceed 1. With the choice (B), the optimality of the TGM with $A_{n_0} = T_{n_0}(f_0)$ has been proven in [27]. With the choice (C), for every $t \geqslant 0$, the cutting matrix $K_{i+1}^i[t]$ coincides with the submatrix of K_{i+1}^i obtained by deleting its first and last t rows with $t = b-1$, where b is the degree of p_i that is equal to the degree of p_0 for $i = 0, \ldots, m-1$ (according to Propositions 4.5 and 5.2, at each level the order of the zeros of f_i is preserved, and therefore the degree of p_i can be maintained constant). This projector is employed in order to preserve the exact Toeplitz structure at each subsequent level of projection.

It is possible to preserve the exact Toeplitz structure at each level, cutting less information. In this paper we propose a different choice, i.e.,

(D) $P_{i+1}^i = K_{i+1}^i \{t\} T_{n_i}(p_i)$,

where t is defined again as the degree of p_0 minus 1 (we remind the reader that the degree of p_i is constant with respect to i), while

$$K_{i+1}^i\{t\} = \left[\; 0_{n_{i+1}-t}^t \mid K_{n_{i+1}-t}^{n_i-2t} \mid 0_{n_{i+1}-t}^t \; \right] \in \mathbf{R}^{(n_{i+1}-t)\times n_i}.$$

Where $0_\alpha^\beta \in \mathbb{R}^{\alpha\times\beta}$ is the null matrix and $K_{n_{i+1}-t}^{n_i-2t} \in \mathbb{R}^{(n_{i+1}-t)\times(n_i-2t)}$ is the usual cutting matrix where we put in evidence the dimensions instead of the recursion levels. We remark that, to apply the MGM recursively, we must start from dimension $n_0 = 2^{k_0} - 1 - 2t$; hence the dimension of problem at each sublevel is $n_i = 2^{k_0-i} - 1 - 2t$. The matrix $K_{i+1}^i\{t\}$ is the cutting matrix that preserves the Toeplitzness at each level cutting the lowest possible level of information. Furthermore, we observe that $K_{i+1}^i[t] = K_{i+1}^i\{2t\}$ for $t \geqslant 1$, and in addition, as can be experimentally verified, the number of iterations required by the MGM to reach a fixed precision is bounded from above by a constant independent of n (optimality). However, the involved constant bound is much higher with the choice (C) than with the choice (D) and this is due to the quantity of information that we lose in the involved choices.

Analogously, in the multilevel case, p_i is a suitable multivariate nonnegative polynomial of partial degrees $t_i + 1$, with $i = 1, \ldots, d$. Let $U_{i+1}^i\{t\} = K_{i+1}^i\{t_1\} \otimes \cdots \otimes K_{i+1}^i\{t_d\}$, we define $P_{i+1}^i = U_{i+1}^i\{t\} T_{n_i}(p_i)$, where $n_i = ((n_i)_1, \ldots, (n_i)_d)$. Therefore, the d-level Toeplitz matrix at MGM recursion level $i+1$ is $T_{n_{i+1}}(f_{i+1}) = P_{i+1}^i T_{n_i}(f_i)(P_{i+1}^i)^H \in \mathbb{R}^{N(n_{i+1})\times N(n_{i+1})}$ for $i = 0, \ldots, m-1$, where each component of n_{i+1} is defined as in the unilevel case.

The definition of the smoothing operators follows the same lines as in section 4.1. In [27] the TGM smoothing property is proved in the Toeplitz case and, by Remark 2.4, we can extend the same property to MGM.

7. Numerical experiments. In this section we present a wide numerical experimentation both in monodimensional and bidimensional cases. We stress that in both

TABLE 7.1
Post-smoother strategies: (a, b, c), where a = iterations of Richardson with $\omega_i = 1/\max(f_i)$, b = iterations of Richardson with $\omega_i = 2/\max(f_i)$, c = iterations of CG; $i = 0, \ldots, m-1$.

	(1, 0, 0)	(2, 0, 0)	(4, 0, 0)	(1, 1, 0)	(1, 0, 1)
$\rho(MGM_0)$	0.75	0.5625	0.3164	0.375	nonstationary
# iterations	83	42	21	25	17

situations we obtain similar results (iteration number independent of problem dimension), so our algorithm performances are not worse for multidimensional problems. In particular, this property is preserved also for the generalization of our algorithm to multilevel Toeplitz systems proposed in section 6.

In what follows, the initial guess is $\mathbf{x}^{(0)} = \mathbf{0}$, the vector $\mathbf{b}$ is calculated from the exact solution $\mathbf{x}_i = i/n$, $i = 1, \ldots, n$. The operations are executed in double precision and the termination condition is $\|\mathbf{r}_0^{(k)}\|_2 \leqslant \varepsilon \|\mathbf{b}\|_2$, where $\varepsilon = 10^{-11}$ in the monodimensional case and $\varepsilon = 10^{-7}$ in the bidimensional case, since the high condition number does not allow a more accurate solution even in double precision. Concerning bidimensional problems, for simplicity, we consider the same dimensions in both directions. In the monodimensional V-cycle algorithm, the system at the coarsest level has dimension $(2^3 - 1) \times (2^3 - 1)$, while in the twodimensional case the size is $(2^3 - 1)^2 \times (2^3 - 1)^2$.

7.1. On the smoother choice. Our theoretical analysis of convergence and optimality is done for only one iteration of a Richardson post-smoother with best parameter $\omega_i = 1/\max(f_i)$ for $i = 0, \ldots, m-1$. Obviously, by increasing the number of iterations of the post-smoother or by adding a pre-smoother, the MGM converges more rapidly. In Table 7.1 we report the MGM spectral radius and the number of iterations when varying the post-smoother strategy. The problem dimension is not reported since by the MGM optimality the spectral radius and the number of iterations does not change for different dimensions. From this table we can see as the MGM spectral radius decreases, the number of iterations required by the method about halves when we double the number of iterations of the smoother. The latter behavior stresses the strength of our MGM, since, by doubling the number of smoothing steps, the overall cost of a single V-cycle iteration is slightly less than doubled. Furthermore, from Table 7.1 we observe that the use of a multi-iterative strategy (see [28]) allows one to increase the MGM convergence speed: here for multi-iterative strategy we mean a fast iterative solver obtained by the combination of possibly slow basic iterations but with spectral complementary behavior. Indeed, one step of post-smoother with Richardson and $\omega_i = 1/\max(f_i)$ and one with $\omega_i = 2/\max(f_i)$, $i = 0, \ldots, m-1$, lead to a V-cycle iteration which is of the same cost as the one with two iterations of post-smoother with Richardson and $\omega_i = 1/\max(f_i)$ for $i = 0, \ldots, m-1$: however the number of iterations for reaching a given accuracy is roughly halved. According to this strategy, using the CG (not a stationary method!), the number of iterations is further reduced as reported in the last column of Table 7.1. We notice that the application of a constant number of CG steps is a nonstationary iteration which reduces to a specific Richardson method with varying parameter when we have only one CG step. The important observation is that both Richardson with $\omega_i = 2/\max(f_i)$ and one step (or a few steps) of the CG method are not smoothers but, according to the terminology of the multi-iterative methods, re intermediate (or residual) iterations. In actuality, in a V-cycle, the smoother "well approximates" the solution in the subspace

where the coefficient matrix is well conditioned, the coarse grid correction (CGC) "well approximates" the solution in the ill-conditioned subspace (if the projector is properly chosen) and the intermediate or residual iteration takes care of the possible subspace where both the smoother and the CGC iterations failed to be effective: it is the spectral complementarity of these basic iterations that makes the whole multi-iterative procedure fast (see [28]). In the specific case of a V-cycle, we observe that the action of the smoother and CGC is enough for obtaining an asymptotically optimal method and therefore the role of the intermediate iteration amounts to accelerating the global convergence speed (see also [31]).

According to the previous reasoning, in the following, when not differently specified, we have used one iteration of relaxed Richardson method with weight $\|f_i\|_\infty^{-1}$ as pre-smoother (a real smoother!) and one CG iteration as post-smoother (a residual iteration). Notice that the latter V-cycle has the same convergence features (see [31]) of a V-cycle with two steps of post-smoothing (one step of Richardson with weight $\|f_i\|_\infty^{-1}$ and one step of CG): we stress that the combination of the former two basic iterations is a smoother, i.e., it satisfies the smoothing property in accordance with Proposition 4.1.

7.2. Elliptic PDEs. Let us consider a d-dimensional problem on the rectangular domain $\Omega = [0,1]^d$:

$$\begin{cases} (-1)^q \sum_{i=1}^{d} \dfrac{\partial^q}{\partial x_i^q}\left(a(x)\dfrac{\partial^q}{\partial x_i^q}u(x)\right) = g(x), \qquad x \in \Omega,\ q \geq 1, \\ \text{homogeneous B.C. on } \partial\Omega \end{cases} \tag{7.1}$$

where $x = (x_1, \ldots, x_d)$, when discretized on a uniform grid of $n = (n_1, \ldots, n_d)$ subintervals using centered finite difference of minimal precision order 2, it leads to a multilevel band $N(n) \times N(n)$ linear system $A_n\mathbf{y} = \mathbf{b}$, that does not belong to the multilevel Toeplitz class unless $a(x)$ is a constant function. In that case $A_n = T_n(f^{(q)})$, where

$$f^{(q)}(x) = \sum_{i=1}^{d} \left[2 - 2\cos(x_i)\right]^q, \tag{7.2}$$

from the condition (4.6) and its generalization to the multidimensional case, we can choose $p_i = p^{(q)}(x)$ with

$$p^{(q)}(x) = \prod_{j=1}^{d} \left[2 + 2\cos(x_j)\right]^q$$

which allows us to obtain the optimality of our MGM when applied to a linear system $\tau_n(f^{(q)})\mathbf{y} = \mathbf{b}$.

For the τ algebra, Table 7.2 shows the number of iterations of our MGM when increasing the dimension n both in the monodimensional and bidimensional case ($n = (n_1, n_2)$). Concerning the circulant algebra, as already stressed, $\mathcal{C}_n(f^{(q)})$ is singular because $f^{(q)}$ vanishes at the origin. Therefore, we solve the system $\widetilde{\mathcal{C}}_n(f^{(q)})\mathbf{y} = \mathbf{b}$, where $\widetilde{\mathcal{C}}_n(f^{(q)})$ is the stabilized version of $\mathcal{C}_n(f^{(q)})$ defined in (1.5). Table 7.3 shows the number of iterations of our MGM applied to these systems in the τ algebra case.

We remark that our MGM shows an optimal behavior also in the multilevel case, and indeed a theoretic extension of the result reported in section 4 to the multidimensional context is reported in [2].

TABLE 7.2
Tau case: Number of iterations for increasing dimensions $N(n)$ both in the monodimensional case ($N(n) = n$) and in the bidimensional case ($N(n) = n_1 n_2$, $n_1 = n_2$).

1D (monodimensional)				2D (bidimensional)			
n	# iterations			$n_1 \cdot n_2$	# iterations		
	$f^{(1)}$	$f^{(2)}$	$f^{(3)}$		$f^{(1)}$	$f^{(2)}$	$f^{(3)}$
2^7-1	14	17	33	$(2^6-1)^2$	11	20	37
2^8-1	14	17	33	$(2^7-1)^2$	11	20	37
2^9-1	14	17	33	$(2^8-1)^2$	10	20	37
$2^{10}-1$	15	17	33	$(2^9-1)^2$	10	20	36

TABLE 7.3
Circulant case with stabilization: Number of iterations for increasing dimensions $N(n)$ both in the monodimensional case ($N(n) = n$) and in the bidimensional case ($N(n) = n_1 n_2$, $n_1 = n_2$).

1D (monodimensional)				2D (bidimensional)			
n	# iterations			$n_1 \cdot n_2$	# iterations		
	$f^{(1)}$	$f^{(2)}$	$f^{(3)}$		$f^{(1)}$	$f^{(2)}$	$f^{(3)}$
2^7	13	17	31	$(2^6)^2$	10	19	34
2^8	14	17	31	$(2^7)^2$	10	19	34
2^9	14	17	31	$(2^8)^2$	10	19	34
2^{10}	14	17	31	$(2^9)^2$	10	19	34

Through the procedure described in section 6 we can also directly solve the system $T_n(f^{(q)})\mathbf{x} = \mathbf{b}$: we must only take care to define the correct dimension to allow the MGM recursive application. In Table 7.4 the degree of p_i is $\lceil (q+1)/2 \rceil$ instead of q (refer to section 8), since the columns number (i.e., information) deleted from our algorithm is proportional to the degree of p_i. Furthermore, to recover a practically optimal behavior we perform $\nu_i = 2 + i$ iterations of pre-smoother and $\nu_i = 2 + i$ iterations of post-smoother at level i (as proposed in [31]). According to the definition of the cutting matrix $K_{i+1}^{i}\{t\}$, to apply recursively the MGM, the dimension of the ith projected system is $2^r - \xi$, where $r \in \mathbb{N}$ and $\xi = 2\lceil (q+1)/2 \rceil - 1$. From Table 7.4 we observe that our MGM shows again a practically optimal behavior both in monodimensional and bidimensional Toeplitz cases and, moreover, we stress that the cost of every MGM iteration with $\nu_i = 2 + i$ is still linear as the size $N(n)$ of the coefficient matrix (see the analysis of the computational cost in [31]).

7.3. Independency from the spectral decomposition of the solution. For our experimentation so far, we obtained the data vector $\mathbf{b}$ from the exact solution $\mathbf{x}_i = i/n$, $i = 1, \ldots, n$ (initial solution $\mathbf{x}^{(0)} = \mathbf{0} \in \mathbb{R}^n$). Here we emphasize that the behavior of our algorithm does not depend on the particular spectral decomposition of the exact solution $\mathbf{x}$. We take four different types of solution where the coefficient matrix of the system is $\tau_n(f^{(2)})$ and in Table 7.5 we report the iteration number required by the MGM to converge. From this table we observe a similar behavior for every different type of solution stressing the robustness of our algorithm.

7.4. Zero not in the origin. We present an example where the generating function f_0 does not vanish at the origin. More explicitly, the symbol

$$f_0(x) = (1 - \cos(x-1))(1 - \cos(x+1)) = (\cos(1) - \cos(x))^2$$

TABLE 7.4
Toeplitz case: Number of iterations for increasing dimensions $N(n)$ both in the monodimensional case ($N(n) = n$) and in the bidimensional case ($N(n) = n_1 n_2$, $n_1 = n_2$) with $\xi = 2\lceil (q+1)/2 \rceil - 1$, $q = 1, 2, 3$.

1D (monodimensional)				2D (bidimensional)			
n	# iterations			$n_1 \cdot n_2$	# iterations		
	$f^{(1)}$	$f^{(2)}$	$f^{(3)}$		$f^{(1)}$	$f^{(2)}$	$f^{(3)}$
$2^7 - \xi$	9	41	53	$(2^6 - \xi)^2$	6	24	33
$2^8 - \xi$	9	44	54	$(2^7 - \xi)^2$	6	26	33
$2^9 - \xi$	10	47	54	$(2^8 - \xi)^2$	6	27	33
$2^{10} - \xi$	9	48	55	$(2^9 - \xi)^2$	6	29	33

TABLE 7.5
Different type of solution: Number of iterations for increasing dimension n for $\tau_n(f^{(2)})$.

n	# iterations for $\mathbf{x}_i =$			
	$\frac{i}{n}$	$(-1)^i$	$\cos\left(\frac{2i\pi}{n}\right)$	1
$2^7 - 1$	17	15	17	17
$2^8 - 1$	17	14	17	17
$2^9 - 1$	17	14	17	17
$2^{10} - 1$	17	14	17	17

is even and for $x \in [0, 2\pi)$ vanishes at 1 and $2\pi - 1$ with order 2. For simplicity, we consider only the monodimensional τ algebra case, but the same considerations hold in the Circulant and Hartley algebras as well. According to (4.3), choosing

$$p_0(x) = (\cos(1) + \cos(x))^2$$

we have an optimal MGM for $\tau_n(f_0)$. Fixing $x_0^{(1)} = 1$ and $x_0^{(2)} = 2\pi - 1$, the position of the new zeros $x_i^{(k)}$ of f_i, for $i = 0, \ldots, m-1$ with $k = 1, 2$, moves according to Proposition 4.5 and then the functions p_i change at each level i. By applying the MGM, since we have two zeros of order two, we strengthen the smoothers by performing two iterations of pre-smoother and post-smoother. In Table 7.6 we report the number of iterations required for convergence, which is practically constant with regard to the dimension n, showing an optimal behavior in this case, too.

7.5. Image restoration problems. In the restoration of blurred images with Dirichlet boundary conditions we solve a system with coefficient matrix $T_n(f)$, where $f(x_1, x_2)$ is small and even indefinite when x_1 and x_2 approach π (see also [6, 18]). Let $\mathcal{S}$ be the true image (for instance a "satellite") and let us consider the blurred image

$$S = T_n(\psi(x_1, x_2)[4 + 2\cos(x_1) + 2\cos(x_2)]^3)\mathcal{S}, \tag{7.3}$$

where the matrix $T_n(\psi(x_1, x_2)[4 + 2\cos(x_1) + 2\cos(x_2)]^3)$ represents the compactly supported and spatially invariant "blurring operator." Here $[4+2\cos(x_1)+2\cos(x_2)]^3$ has a zero at (π, π) of order 6 and $\psi(x_1, x_2)$ is a strictly positive polynomial with nonnegative Fourier coefficients: in this way the Fourier coefficients of $\psi(x_1, x_2)[4 +$

TABLE 7.6
Zero not in the origin: Number of iterations increasing the dimension for $\tau_n((\cos(1)-\cos(x))^2)$.

dimension	2^7-1	2^8-1	2^9-1	$2^{10}-1$
# iterations	18	27	28	26

TABLE 7.7
"Satellite" restoration: Error behavior in $\|\cdot\|_2$.

# iterations	1	10	20	30	42
error norm	8.271856E-01	4.522643E-03	4.490511E-04	5.478008E-05	4.781925E-06

$2\cos(x_1)+2\cos(x_2)]^3$ are nonnegative as reported in the following mask:

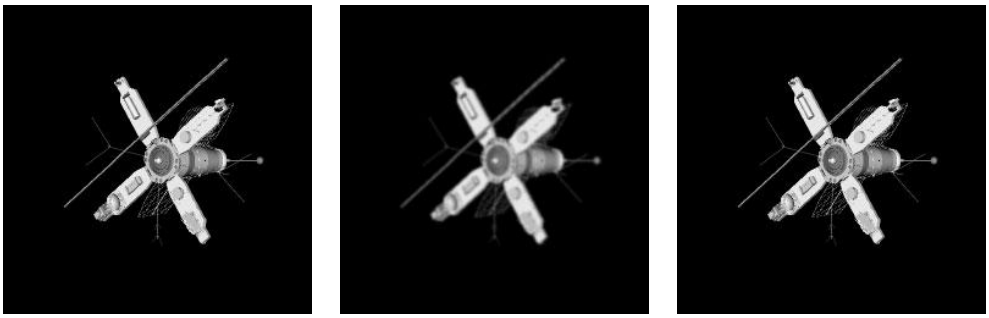
0	0	0	0	0.0002	0	0	0	0
0	0	0	0.0007	0.0033	0.0007	0	0	0
0	0	0.0010	0.0098	0.0260	0.0098	0.0010	0	0
0	0.0007	0.0098	0.0508	0.1022	0.0508	0.0098	0.0007	0
0.0002	0.0033	0.0260	0.1022	0.1829	0.1022	0.0260	0.0033	0.0002
0	0.0007	0.0098	0.0508	0.1022	0.0508	0.0098	0.0007	0
0	0	0.0010	0.0098	0.0260	0.0098	0.0010	0	0
0	0	0	0.0007	0.0033	0.0007	0	0	0
0	0	0	0	0.0002	0	0	0	0

Therefore, the associated Toeplitz sequence is asymptotically very ill-conditioned ($\sim [N(n)]^3$) and, despite this bad spectral behavior, the proposed multigrid method is optimal as emphasized by the linear convergence reported in Table 7.7. The considered choice is made in such a way that the resulting blur operator is a band approximation of the classical Gaussian blur whose Fourier coefficients are positive, symmetric and decay exponentially and whose generating function is close to zero in a neighborhood of (π,π) and is positive elsewhere. Furthermore, the presence of the term $\psi(x_1,x_2)>0$ leads to a larger bandwidth so that the resulting blurring effect is more realistic.

As in the monodimensional case described in section 4.5, in the multidimensional case also, our discretized integral problem is projected at the lower level into a discretized differential problem, so that the optimal behavior holds as shown in section 7.2 and Table 7.7. We consider the blurred image without noise and we solve the system (7.3) with the same smoother choice performed in section 7.2 for the Toeplitz case.

We stress that the regularization is not necessary since the image and the point spread function (PSF) are noise free and the conditioning of the blur operator is only polynomial with the size of the matrix. In Figure 7.1 we report the sequence of "satellite" image, the true image $\mathcal{S}$, the blurred image S, and the restored image with our MGM after 42 iterations.

Finally, we remark that in the case of noise the regularized systems $(T_n(f)+\mu I)\mathcal{S}=S$ with $\mu>0$ (see [6]) have a better conditioning than in the case of $\mu=0$: therefore, our multigrid procedure, which is optimal for $\mu=0$, will be robust since the number of iterations will be bounded by a constant independent both of $N(n)$ and of $\mu>0$. In the first line of Table 7.8 we report the number of iterations for the restoration of the blurred satellite affected by 2% of noise with varying μ and with our projector. The second line of that table is obtained by using the same V-cycle with the same smoothers and with the classical projector used in the PDEs context [17]



True Image (dim: 253×253*).* *Blurred Image.* *Restored Image after* 42 *iterations.*

FIG. 7.1. *Sequence of satellite images.*

TABLE 7.8
Number of iterations for the satellite restoration with 2% *of noise with varying* μ*.*

μ	10^{-1}	10^{-2}	10^{-3}	10^{-4}
$p(x) = (2 - 2\cos(x_1))^2(2 - 2\cos(x_2))^2$ our projector	7	28	67	94
$p(x) = (2 + 2\cos(x_1))(2 + 2\cos(x_2))$ linear interpolation	7	32	216	1806

and by Huckle et al. [18] and R. Chan, T. Chan, and W. Wan [6] in image restoration: it is evident that our choice improves the convergence behavior substantially by maintaining the same computational cost. However, in the numerics with Gaussian blur in [6, 18] the authors obtained reasonably good results by applying the "wrong" prolongation operator: the reason is that, as smoothers, they used very sophisticated and costly solvers like PCG and FGMRES (flexible GMRES) with cosine/circulant preconditioners. Therefore the success of the whole procedure is mainly due to these auxiliary solvers which are reasonably effective on their own. Future work should try to combine their approach (with sophisticated smoothers) and the "correct" prolongation operators indicated in the present paper.

8. Concluding remarks, open problems, and future work. In this paper we have proposed a proof technique (based on matrix inequalities and on the Perron–Frobenius theorem) which has been successful for a rigorous convergence analysis of the V-cycle procedure when applied to unilevel linear systems from algebras. We have also presented some algorithmic proposals for multilevel and Toeplitz structures: the numerical results (on discretized differential and integral problems) indicate an optimal convergence rate of our V-cycle procedures, but still we have to provide a theoretical analysis in the multilevel and Toeplitz settings.

Therefore future work should include the following directions:

- Toeplitz extension of the theory (for the TGM this has been done in [27]);
- multilevel extension of the theory (for matrix algebras see [2] while for multilevel Toeplitz structures only the TGM analysis is available [27]);
- multiple zero case (the TGM analysis and the numerical results are available [7, 15, 16, 18, 27, 31]: the MGM theory should be easy but tedious following the approach in the present paper).

Moreover, from an experimental viewpoint it is evident that conditions (2.15) are

sufficient for the level independency and for the TGM optimality but they are not enough for the MGM optimality (see section 3). We have proven conditions (4.2) to be sufficient for the MGM optimality. However, from our numerics and from [31], we know that conditions (4.2) can be replaced by

$$\begin{aligned} \lim_{x \to x^0} \frac{p_i^2(\pi + x)}{f_i(x)} &= 0, \\ p_i^2(x) + p_i^2(\pi + x) &> 0 \qquad \forall\, x \end{aligned}$$

by preserving the MGM optimal convergence rate (the latter are much weaker than (4.2) and just a little bit stronger than (2.15)!). Future work should also try to answer the previous question.

REFERENCES

[1] A. Aricò, *Metodi Multigrid in Algebre Trigonometriche*, BD thesis in Mathematics, University of Pisa, Pisa, Italy, 2002.

[2] A. Aricò and M. Donatelli, *V-Cycle Optimal Convergence for Multilevel Structures*, Numerische Mathematik, submitted, 2003.

[3] D. Bini and M. Capovani, *Spectral and computational properties of band symmetric Toeplitz matrices*, Linear Algebra Appl., 52/53 (1983), pp. 99–126.

[4] D. Bini and P. Favati, *On a matrix algebra related to the discrete Hartley transform*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 500–507.

[5] W. Briggs, V. Henson, and S. McCormick, *A Multigrid Tutorial*, 2nd ed., SIAM, Philadelphia, 2000.

[6] R.H. Chan, T.F. Chan, and W. Wan, *Multigrid for differential-convolution problems arising from image processing*, in Scientific Computing, G. Golub, S.H. Lui, F. Luk, and R. Plemmons, eds., Springer-Verlag, Singapore, 1999, pp. 58–72.

[7] R.H. Chan, Q. Chang, and H. Sun, *Multigrid method for ill-conditioned symmetric Toeplitz systems*, SIAM J. Sci. Comput., 19 (1998), pp. 516–529.

[8] R.H. Chan, M. Donatelli, S. Serra-Capizzano, and C. Tablino Possio, *Application of multigrid techniques to image restoration problems*, Research Report CUHK-2002-14 (254), Department of Mathematics, The Chinese University of Hong Kong.

[9] R.H. Chan and M. Ng, *Conjugate gradient methods for Toeplitz systems*, SIAM Rev., 38 (1996), pp. 427–482.

[10] R.H. Chan and G. Strang, *Toeplitz equations by conjugate gradients with circulant preconditioner*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 104–119.

[11] T.F. Chan, *An optimal circulant preconditioner for Toeplitz systems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 766–771.

[12] F. Di Benedetto, *Analysis of preconditioning techniques for ill-conditioned Toeplitz matrices*, SIAM J. Sci. Comput., 16 (1995), pp. 682–697.

[13] F. Di Benedetto, G. Fiorentino, and S. Serra-Capizzano, *C.G. preconditioning for Toeplitz matrices*, Comput. Math. Appl., 25 (1993), pp. 35–45.

[14] M. Donatelli, *Metodi Multigrid per Sistemi Lineari Strutturati Ed Applicazioni*, BD thesis in Computer Science, University of Florence, 2002.

[15] G. Fiorentino and S. Serra-Capizzano, *Multigrid methods for Toeplitz matrices*, Calcolo, 28 (1991), pp. 283–305.

[16] G. Fiorentino and S. Serra-Capizzano, *Multigrid methods for symmetric positive definite block Toeplitz matrices with nonnegative generating functions*, SIAM J. Sci. Comput., 17 (1996), pp. 1068–1081.

[17] W. Hackbush, *Multi-grid Methods and Applications*, Springer-Verlag, New York, 1979.

[18] T. Huckle and J. Staudacher, *Multigrid preconditioning and Toeplitz matrices*, Electron. Trans. Numer. Anal., 13 (2002), pp. 81–105.

[19] D.G. Luenberger, *Introduction to Dynamic Systems (Theory, Models & Applications)*, John Wiley & Sons Inc., New York, 1979.

[20] M. Ng, R. Chan and W.C. Tang *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM J. Sci. Comput., 21 (1999), pp. 851–866.

[21] D. Noutsos, S. Serra-Capizzano, and P. Vassalos, *Spectral equivalence and matrix algebra preconditioners for multilevel Toeplitz systems: A negative result*, in Fast Algorithms for

Structured Matrices: Theory and Applications, V. Olshevsky, ed., Contemp. Math. 323, AMS, Providence, RI, 2003, pp. 313–322.

[22] D. Noutsos, S. Serra-Capizzano, and P. Vassalos, *Matrix algebra preconditioners for multilevel Toeplitz systems do not insure an optimal convergence rate*, V. Pan ed., Theoret. Comp. Sci., to appear.

[23] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, Frontiers Appl. Math. 3, S.F. McCormick, ed., SIAM, Philadelphia, 1987, pp. 73–130.

[24] S. Serra-Capizzano, *Proprietà Algebriche e Computazionali di Matrici di Toeplitz e Metodi Multigrid*, BD thesis in Computer Science, University of Pisa, 1990.

[25] S. Serra-Capizzano, *Toeplitz preconditioners constructed from linear approximation processes*,SIAM J. Matrix Anal. Appl., 20 (1998), pp. 446–465.

[26] S. Serra-Capizzano, *Matrix algebra preconditioners for multilevel Toeplitz matrices are not superlinear*, Linear Algebra Appl., 343/344 (2002), pp. 303–319.

[27] S. Serra-Capizzano, *Convergence analysis of two-grid methods for elliptic Toeplitz and PDEs matrix-sequences*, Numer. Math., 92 (2002), pp. 433–465.

[28] S. Serra-Capizzano, *Multi-iterative methods*, Comput. Math. Appl., 26 (1993), pp. 65–87.

[29] S. Serra-Capizzano, *A note on anti-reflective boundary conditions and fast deblurring models*, SIAM J. Sci. Comput., 25 (2003), pp. 1307–1325.

[30] S. Serra-Capizzano and C. Tablino Possio, *Spectral and structural analysis of high precision finite difference matrices for elliptic operators*, Linear Algebra Appl., 293 (1999), pp. 85–131.

[31] S. Serra-Capizzano and C. Tablino Possio, *Multigrid methods for multilevel circulant matrices*, SIAM J. Sci. Comput., to appear.

[32] S. Serra-Capizzano and E. Tyrtyshnikov, *Any circulant-like preconditioner for multilevel matrices is not superlinear*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 431–439.

[33] S. Serra-Capizzano and E. Tyrtyshnikov, *How to prove that a preconditioner cannot be superlinear*, Math. Comp., 72 (2003), pp. 1305–1316.

[34] E. Tyrtyshnikov, *Circulant preconditioners with unbounded inverse*, Linear Algebra Appl., 216 (1995), pp. 1–23.

[35] E. Tyrtyshnikov, *A unifying approach to some old and new theorems on distribution and clustering*, Linear Algebra Appl., 232 (1996), pp. 1–43.

[36] R.S. Varga, *Matrix Iterative Analysis*, Prentice Hall, Englewood Cliffs, NJ, 1962.

[37] H. Yserentant, *Old and new convergence proofs for multigrid methods*, Acta Numer., 2 (1993), pp. 285–326.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 1, pp. 215–232

TRIDIAGONAL-DIAGONAL REDUCTION OF SYMMETRIC INDEFINITE PAIRS*

FRANÇOISE TISSEUR†

Abstract. We consider the reduction of a symmetric indefinite matrix pair (A, B), with B nonsingular, to tridiagonal-diagonal form by congruence transformations. This is an important reduction in solving polynomial eigenvalue problems with symmetric coefficient matrices and in frequency response computations. The pair is first reduced to symmetric-diagonal form. We describe three methods for reducing the symmetric-diagonal pair to tridiagonal-diagonal form. Two of them employ more stable versions of Brebner and Grad's pseudosymmetric Givens and pseudosymmetric Householder reductions, while the third is new and based on a combination of Householder reflectors and hyperbolic rotations. We prove an optimality condition for the transformations used in the third reduction. We present numerical experiments that compare the different approaches and show improvements over Brebner and Grad's reductions.

Key words. symmetric indefinite generalized eigenvalue problem, tridiagonalization, hyperbolic rotation, unified rotation, hyperbolic Householder reflector

AMS subject classifications. 65F15, 65F30

DOI. 10.1137/S0895479802414783

1. Introduction.

Motivation for this work comes from the symmetric polynomial eigenvalue problem (PEP)

$$(\lambda^m A_m + \lambda^{m-1} A_{m-1} + \cdots + A_0)u = 0, \tag{1.1}$$

where the A_i, $i = 0\colon m$, are $n \times n$ symmetric matrices. λ is called an eigenvalue and $u \neq 0$ is the corresponding right eigenvector. The standard way of dealing with the PEP in practice is to reformulate it as a generalized eigenvalue problem (GEP)

$$Ax = \lambda Bx, \tag{1.2}$$

of size mn. This process is called linearization, as the GEP is linear in λ. Symmetry in the problem is maintained with an appropriate choice of linearization. For example, we can take

$$A = \begin{bmatrix} 0 & \cdots & \cdots & 0 & A_0 \\ \vdots & & & A_0 & A_1 \\ \vdots & & & \vdots & \vdots \\ 0 & A_0 & & & A_{m-2} \\ A_0 & A_1 & \cdots & A_{m-2} & A_{m-1} \end{bmatrix}, \quad B = \begin{bmatrix} 0 & \cdots & 0 & A_0 & 0 \\ \vdots & & A_0 & A_1 & \vdots \\ & & & & \\ 0 & A_0 & & \vdots & \vdots \\ A_0 & A_1 & \cdots & A_{m-2} & 0 \\ 0 & \cdots & \cdots & 0 & -A_m \end{bmatrix}$$

and $x = [u^T, \lambda u^T, \ldots, \lambda^{m-1} u^T]^T$. The resulting A and B are symmetric but not definite, and in general the pair (A, B) is indefinite.

*Received by the editors September 16, 2002; accepted for publication (in revised form) by I. S. Dhillon November 14, 2003; published electronically September 14, 2004. This work was supported by Engineering and Physical Sciences Research Council grant GR/R45079 and Nuffield Foundation grant NAL/00216/G.

http://www.siam.org/journals/simax/26-1/41478.html

†Department of Mathematics, University of Manchester, Manchester, M13 9PL, UK (ftisseur@ma.man.ac.uk, http://www.ma.man.ac.uk/~ftisseur/).

The first step in most eigensystem computations is the reduction of the coefficient matrices, in a finite number of operations, to a simple form. Only then is an iterative procedure applied. A symmetric indefinite pair (A, B) can be reduced to Hessenberg-triangular form and the resulting generalized eigenvalue problem solved by the QZ algorithm. This approach is numerically stable, but unfortunately the reduction to Hessenberg-triangular form destroys the symmetry. Moreover, in finite precision arithmetic there is no guarantee that the set of left and right eigenvectors computed via the QZ algorithm will coincide, a property possessed by GEPs with real symmetric matrices. Also, by preserving symmetry, storage and computational costs can be reduced.

The tridiagonal-diagonal reduction of a pair (A, B) is the most compact form we can obtain in a finite number of steps. Such reductions have been proposed by Brebner and Grad [5] and by Zurmühl and Falk [26] for nonsingular B. They require nonorthogonal transformations and can be unstable. Once (A, B) is reduced to tridiagonal-diagonal form the eigenvalues and eigenvectors can be obtained by applying, for example, an HR iteration or associated iterations [5], [6], [16], [25], Uhlig's DQR algorithm [24], or, if one is interested in the eigenvalues only, Aberth's method can be used in an efficient way [1]. A robust tridiagonal-diagonal reduction is therefore of prime importance before one can consider using any of the methods cited above. We note that Garvey et al. [8] have considered a less compact form that allows the second matrix to be in tridiagonal form. One feature of their approach is that no assumption is made on the nonsingularity of the two matrices. The simultaneous tridiagonalization is convenient if one needs to solve linear systems of the form $(A - \omega B)x = b$ for many values of ω, as is required in frequency response computations [8], but it is less attractive than the tridiagonal-diagonal form for eigenvalue computations.

Three different tridiagonal-diagonal reductions for indefinite pairs (A, B) with B nonsingular are described in this paper. They all consist of two stages. The first, common to all, is the reduction of the symmetric indefinite pair (A, B) to symmetric-diagonal form (C, J) with the aid of a block LDL^T factorization of B. During the second stage, C is tridiagonalized using a sequence of congruence transformations that preserve the diagonal form of the second matrix J. Each of the three reductions proposed in this paper uses different types of transformations. These transformations are not necessarily orthogonal, so they may be unstable in finite precision arithmetic. We describe several techniques that can be used to make them more robust and to improve stability during the reduction process: in particular, pivoting and zeroing strategies in order to minimize the condition numbers of the transformations, and mixed application of hyperbolic rotations.

The paper is organized as follows. Section 2 sets up notations and definitions. It is shown that if the tridiagonal-diagonal reduction exists, it is determined up to signs by the first column of the transformation matrix. Section 3 describes the first stage of the reduction, that is, the reduction of (A, B) to symmetric-diagonal form (C, J). The description is accompanied by an error analysis. The second stage of the reduction is described in section 4. Three algorithms are proposed. The first two are an improvement over Brebner and Grad's pseudosymmetric Givens and pseudosymmetric Householder methods [5]. The third algorithm is based on transformations used to compute hyperbolic QR factorizations in indefinite least square problems [3]. Numerical comparisons of these algorithms and comparisons to Brebner and Grad's reductions are given in the last section.

2. Background material. Unless otherwise specified, $\|\cdot\|$ denotes the 2-norm. We denote by $\mathrm{diag}_q^n(\pm 1)$ the set of all $n \times n$ diagonal matrices with q diagonal elements

equal to 1 and $n-q$ equal to -1. A matrix $J \in \operatorname{diag}_q^n(\pm 1)$ for some q is called a *signature* matrix.

Let J, $\widetilde{J} \in \operatorname{diag}_q^n(\pm 1)$. A matrix $H \in \mathbb{R}^{n\times n}$ is said to be $(J,\widetilde{J})$-*orthogonal* if $H^T J H = \widetilde{J}$. Note that $(J,\widetilde{J})$-orthogonal matrices are sometimes called $(J,\widetilde{J})$-hyperexchange or $(J,\widetilde{J})$-hypernormal matrices in the signal processing literature [17].

We recall that a tridiagonal matrix is *unreduced* if none of its next-to-diagonal elements (that is, the elements on the first subdiagonal and the first superdiagonal) is zero.

The following result is related to the implicit Q theorem [11]. A more general form can be found in [18, Thm. 2.2].

THEOREM 2.1. *If $C \in \mathbb{R}^{n\times n}$ admits a representation of the form*

$$Q^T C Q = T, \tag{2.1}$$

where T is unreduced tridiagonal and Q is $(J,\widetilde{J})$-orthogonal, then the columns of Q and the next-to-diagonal elements of T are determined up to signs by the first (or last) column of Q.

We give the proof since we need to refer to it later in the text. This is a constructive proof that describes a Lanczos process.

Proof. Let $\widetilde{J} = \operatorname{diag}(\widetilde{\sigma}_i)$, $\widetilde{\sigma}_i = \pm 1$, $i = 1\colon n$, and

$$T = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \ddots & \ddots & \\ & & \ddots & \alpha_{n-1} & \beta_n \\ & & & \beta_n & \alpha_n \end{bmatrix}.$$

We assume that q_1 is given and normalized such that $\widetilde{\sigma}_1 = q_1^T J q_1$. This yields

$$\alpha_1 = q_1^T C q_1.$$

Using the $(J,\widetilde{J})$-orthogonality of Q, equation (2.1) can be rewritten as

$$JCQ = Q\widetilde{J}T. \tag{2.2}$$

Equating the first column on each side of (2.2) gives

$$p_1 := JCq_1 - \alpha_1\widetilde{\sigma}_1 q_1 = \beta_2\widetilde{\sigma}_2 q_2.$$

From the $(J,\widetilde{J})$-orthogonality of Q we get $\widetilde{\sigma}_2 = \beta_2^{-2} p_1^T J p_1$, which implies

$$\widetilde{\sigma}_2 = \operatorname{sign}(p_1^T J p_1), \quad \beta_2 = \pm\sqrt{|p_1^T J p_1|},$$

so that $q_2 = \widetilde{\sigma}_2\beta_2^{-1} p_1$ is determined up to the sign chosen for β_2. The second diagonal element of T is uniquely determined by

$$\alpha_2 = q_2^T C q_2.$$

Hence, the construction of q_2, α_2, β_2, and $\widetilde{\sigma}_2$ requires just the knowledge of p_1. Now suppose that the first $j < n$ columns of Q and the leading $j \times j$ principal submatrices

of T and $\widetilde{J}$ are known. Then by equating the jth columns on each side of (2.2) we obtain

$$p_j := JCq_j - \widetilde{\sigma}_j\alpha_j q_j - \widetilde{\sigma}_{j-1}\beta_j q_{j-1} = \widetilde{\sigma}_{j+1}\beta_{j+1}q_{j+1}.$$

Using once again the $(J,\widetilde{J})$-orthogonality of Q we have

$$\widetilde{\sigma}_{j+1} = \operatorname{sign}(p_j^T Jp_j), \quad \beta_{j+1} = \pm\sqrt{|p_j^T Jp_j|}. \tag{2.3}$$

Hence

$$q_{j+1} = \widetilde{\sigma}_{j+1}\beta_{j+1}^{-1}p_j, \quad \alpha_{j+1} = q_{j+1}^T Cq_{j+1}. \tag{2.4}$$

Again, β_{j+1} and q_{j+1} are determined up to a sign. By induction on j all columns of Q and all next-to-diagonal elements of T are determined, up to a sign by q_1.

The proof is similar if q_n, the last column of Q is chosen in place of q_1. ∎

For a particular q_1, the proof shows that if, for some $j \le n$, $p_j^T Jp_j = 0$, the reduction breaks down. If $p_j = 0$ then $\beta_{j+1} = 0$. We can carry on the construction with a new q_{j+1} chosen to be J-orthogonal to the previous q_k, $k = 1\colon j$. If $p_j^T Jp_j = 0$ but $p_j \neq 0$ then the breakdown is serious and there is no $(J,\widetilde{J})$-orthogonal matrix Q with this given q_1 that satisfies (2.1). In this case, q_1 is called *exceptional*.

The construction of the quantities q_{j+1}, α_{j+1}, β_{j+1}, and $\widetilde{\sigma}_{j+1}$ in (2.3) and (2.4) corresponds to a modification of the Lanczos process for symmetric matrices and therefore provides a numerical method for the reduction of a symmetric-diagonal pair to tridiagonal-diagonal form. We will instead consider methods based on a finite sequence of unified rotations or unified Householder reflectors or a mix of hyperbolic rotations and Householder reflectors. But before describing the tridiagonalization process we first consider the reduction of the symmetric indefinite pair (A,B) to symmetric-diagonal form.

3. Reduction to symmetric-diagonal form. Since B is indefinite we use a block LDL^T factorization [13, Chap. 11]

$$P^T BP = LDL^T, \tag{3.1}$$

where P is a permutation matrix, L is unit lower triangular and D is diagonal with 1×1 or 2×2 blocks on its diagonal. This factorization costs $n^3/3$ operations plus the cost of determining the permutation matrix. There are several possible choices for P (see [13, sect. 11.1] for a detailed description and stability analysis). We opt for the symmetric rook pivoting strategy [13, sect. 9.1], as it yields a factor L with bounded elements. Let

$$D = X|\Lambda|^{1/2}J|\Lambda|^{1/2}X^T, \quad J \in \operatorname{diag}_q^n(\pm 1), \tag{3.2}$$

be the eigendecomposition of D, where X is orthogonal and Λ is the diagonal matrix of eigenvalues. Note that X has the same structure as D with the 1×1 blocks equal to 1 and the 2×2 blocks can be chosen to be Jacobi rotations of the form

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \quad c^2 + s^2 = 1.$$

The pair (C,J) with

$$C = M^T AM, \quad M = PL^{-T}X|\Lambda|^{-1/2} \tag{3.3}$$

is congruent to (A, B) and is in symmetric-diagonal form.

The following pseudocode constructs C, J, and the transformation matrix M in (3.3). We assume that a function computing a LDL^T factorization with rook pivoting is available. For example, we can use the MATLAB function `ldlt_symm` from Higham's Matrix Computation Toolbox [12].

function $[C, J, M] = \text{sym\_diag}(A, B)$
% Compute C, J, and M so that $M^T(A, B)M = (C, J)$
% is a symmetric-diagonal pair.
Compute the factorization $P^T BP = LDL^T$
$X = I$
for $k = 1 : n - 1$
  if $D(k + 1, k) \neq 0$
    $\tau = 0.5(D(k + 1, k + 1) - D(k, k))/D(k + 1, k)$
    if $\tau \geq 0$
      $t = 1/(\tau + \sqrt{1 + \tau^2})$
    else
      $t = -1/(-\tau + \sqrt{1 + \tau^2})$
    end
    $c = 1/\sqrt{1 + t^2}$, $s = tc$
    $X[k\!:\!k + 1, k\!:\!k + 1] = \left[\begin{smallmatrix} c & s \\ -s & c \end{smallmatrix}\right]$
    $\alpha = D(k, k) - D(k + 1, k)t$
    $\beta = D(k + 1, k + 1) + D(k + 1, k)t$
    $D(k\!:\!k + 1, k\!:\!k + 1) = \left[\begin{smallmatrix} \alpha & 0 \\ 0 & \beta \end{smallmatrix}\right]$
  end
end
$J = \text{sign}(D)$,
$C = |D|^{-1/2} X^T L^{-1} (PAP^T) L^{-T} X |D|^{1/2}$
$M = PL^{-T} X |D|^{-1/2}$

We now give a rounding error analysis of this reduction. We use the standard model of floating point arithmetic [13, sect. 2.2]:

$$fl(x \operatorname{op} y) = (x \operatorname{op} y)(1 + \delta)^{\pm 1}, \qquad |\delta| \leq u, \quad \operatorname{op} = +, -, *, /,$$

where u is the unit roundoff.

Let $\widehat{L}\widehat{D}\widehat{L}^T$ be the computed factorization in (3.1). Using a general result on the stability of block LDL^T factorization [13, Thm. 11.3], we have

$$P^T(B + \Delta B_1)P = \widehat{L}\widehat{D}\widehat{L}^T, \ |\Delta B_1| \leq p(n)\, u(|B| + P|\widehat{L}||\widehat{D}||\widehat{L}^T|P^T) + O(u^2) \tag{3.4}$$

with p a linear polynomial.

Slapničar [22] shows that when a Jacobi rotation is used to compute the decomposition $H = GJG^T$ of a symmetric $H \in \mathbb{R}^{2\times 2}$ and $J \in \text{diag}(\pm 1)$, the computed decomposition $\widehat{G}\widehat{J}\widehat{G}^T$ satisfies

$$\widehat{G}\widehat{J}\widehat{G}^T = H + \Delta H, \quad |\Delta H| \leq \alpha |G||G^T| u,$$

with α a small integer constant. Using this result we obtain for the computed eigendecomposition (3.2)

$$\widehat{X}|\widehat{\Lambda}|^{1/2}\widehat{J}|\widehat{\Lambda}|^{1/2}\widehat{X}^T = \widehat{D} + \Delta\widehat{D}, \qquad |\Delta\widehat{D}| \leq \tilde{\alpha} u |\widehat{X}|\ |\widehat{\Lambda}|\ |\widehat{X}^T| \tag{3.5}$$

with $\tilde{\alpha}$ a small integer constant. Combining (3.4) with (3.5), we have

$$P^T(B+\Delta B)P = \widehat{L}\widehat{X}\ |\widehat{\Lambda}|^{1/2}\widehat{J}|\widehat{\Lambda}|^{1/2}\widehat{X}^T\widehat{L}^T,$$

where

$$|\Delta B| \le p'(n)u(|B| + P|\widehat{L}|\ |\widehat{X}|\ |\widehat{\Lambda}|\ |\widehat{X}^T|\ |\widehat{L}^T|P^T) + O(u^2).$$

This is the best form of bound we could expect. Note that if rook pivoting is used then all the entries of L are bounded by 2.78 [13, sect. 11.1.3].

Using standard results [13] on the componentwise backward error in solving triangular systems and componentwise backward errors in the product of matrices we find, after some algebraic manipulations, that the computed $\widehat{C}$ satisfies

$$\widehat{C} = |\widehat{\Lambda}|^{-1/2}\widehat{X}^T\widehat{L}^{-1}P^T(A+\Delta A)P\widehat{L}^{-T}\widehat{X}|\widehat{\Lambda}|^{-1/2},$$

where

$$\begin{aligned}|\Delta A| \le \gamma_n\Big(&P|\widehat{L}||\widehat{L}^{-1}||A|(I + |\widehat{L}^{-T}||\widehat{L}^T|P^T)\\ &+ P|\widehat{L}||\widehat{X}||\widehat{X}^T||\widehat{L}^{-1}||A||\widehat{L}^{-T}|(I + |\widehat{X}^T||\widehat{X}|)|\widehat{L}^T|P^T\Big)\end{aligned}$$

with $\gamma_n = nu/(1-nu)$. Taking the ∞-norm gives

$$\|\Delta A\|_\infty \le \widetilde{\gamma}_n\kappa_\infty(L)^2\|A\|_\infty, \tag{3.6}$$

with $\widetilde{\gamma}_n = cnu/(1-cnu)$, c being a small integer constant.

This is the same form of normwise backward error result as we obtain for the reduction of a symmetric definite pair (A,B) with B positive definite using a Cholesky decomposition of B [7]. If rook pivoting is used in the block LDL^T factorization then [13, Prob. 8.5]

$$\kappa_\infty(L) = \|L\|_\infty\|L^{-1}\|_\infty \le 3.78^{n-1}\big(1 + 2.78(n-1)\big),$$

and so $\|\Delta A\|_\infty$ in (3.6) is bounded independently of B. For the definite case, if complete pivoting in the Cholesky factorization is used, we have the smaller bound $\kappa_\infty(L) \le n2^{n-1}$.

4. Reduction to tridiagonal-diagonal form. Given a symmetric-diagonal pair (C,J) with $J \in \operatorname{diag}_q^n(\pm 1)$, this section deals with the construction of a nonsingular matrix Q such that

$$Q^TCQ = T, \qquad Q^TJQ = \widetilde{J}, \tag{4.1}$$

with T symmetric tridiagonal and $\widetilde{J} \in \operatorname{diag}_n^q(\pm 1)$. We denote by σ_i and $\widetilde{\sigma}_i$ the ith diagonal element of J and $\widetilde{J}$, respectively.

Brebner and Grad [5] propose two methods: a pseudosymmetric Givens method and a pseudosymmetric Householder method. Both reduce the pseudosymmetric[1] matrix JC to pseudosymmetric tridiagonal form $\widetilde{T} = \widetilde{J}T$ with $\widetilde{J} \in \operatorname{diag}_q^n(\pm 1)$ and T symmetric tridiagonal. Their reduction is equivalent to reducing $C - \lambda J$ to symmetric

[1] A matrix M is pseudosymmetric if $M = NJ$ where $N = N^T$ and $J = \operatorname{diag}(\pm 1)$. Equivalently, MJ (or JM) is symmetric.

tridiagonal-diagonal form $T - \lambda\widetilde{J}$ using a sequence of Givens and hyperbolic transformations or a sequence of hyperbolic Householder transformations. The first two reductions described below are based on similar ideas. They contain several improvements over Brebner and Grad's reductions that make them more stable. The third reduction is new and based on a combination of Householder reflectors and hyperbolic rotations.

4.1. Reduction by unified rotation. The term *unified rotation* was introduced by Bojanczyk, Qiao, and Steinhardt [4]. Unified rotations include both orthogonal and hyperbolic rotations. Given a 2×2 signature matrix $J = \operatorname{diag}(\sigma_1, \sigma_2)$, unified rotations have the form

$$G = \begin{bmatrix} c & \frac{\sigma_2}{\sigma_1}s \\ -s & c \end{bmatrix} \in \mathbb{R}^{2\times 2}, \quad \sigma_1 c^2 + \sigma_2 s^2 = \widetilde{\sigma}_1, \quad \widetilde{\sigma}_1 = \pm 1. \tag{4.2}$$

If we define $\widetilde{\sigma}_2 = \sigma_2\widetilde{\sigma}_1/\sigma_1$ then $G^T J G = \operatorname{diag}(\widetilde{\sigma}_1, \widetilde{\sigma}_2) \equiv \widetilde{J}$, that is, G is $(J, \widetilde{J})$-orthogonal. Thus G is a Givens rotation when $J = \pm I$ and a hyperbolic rotation when $J \neq \pm I$. Hyperbolic rotations are said to be of *type* 1 when $J = \widetilde{J}$ and of *type* 2 when $J = -\widetilde{J}$. Let $x = [x_1, x_2]^T \neq 0$ be such that $x^T J x \neq 0$. Choosing

$$c = x_1/\sqrt{|x^T J x|}, \qquad s = x_2/\sqrt{|x^T J x|}$$

gives $Gx = [\rho, 0]^T$ with $\rho = (\widetilde{\sigma}_1/\sigma_1)\sqrt{|x^T J x|}$ and $\widetilde{\sigma}_1 = \operatorname{sign}(x^T J x)$.

The following pseudocode, inspired by [4, Alg. 2] constructs c and s and guards against the risk of overflow.

```
function [c, s, J̃] = u_rotate(x, J)
% Given x = [x1, x2]^T and J = diag(σ1, σ2), compute c and s defining the
% unified rotation G such that Gx has zero second element and G is
% (J, J̃)-orthogonal.
γ = σ2/σ1, J̃ = J
if x2 = 0
   s = 0, c = 1, return
end
if |x1| = −γ|x2|
   No unified rotation exists—abort.
end
if |x1| > |x2|
   t = x2/x1, τ = 1 + γ t^2
   c = sign(x1)/√τ, s = ct
else
   t = x1/x2, τ = γ + t^2
   s = sign(x2)/√|τ|, c = st
end
if τ < 0, J̃ = −J̃, end
```

Bojanczyk, Brent, and Van Dooren [2] noticed that how hyperbolic rotations are applied to a vector is crucial to the stability of the computation. Consider the computation of $y = Gx$ with $\sigma_2/\sigma_1 = -1$:

$$y_1 = cx_1 - sx_2, \tag{4.3}$$

$$y_2 = -sx_1 + cx_2. \tag{4.4}$$

We call (4.3)–(4.4) the *direct application* of G to a vector x. When $\sigma_1 = \widetilde{\sigma}_1$ (i.e., for hyperbolic rotations of type 1), we compute y_1 from (4.3). Solving (4.3) for x_1 gives

$$x_1 = \frac{y_1}{c} + \frac{s}{c}x_2, \tag{4.5}$$

which allows (4.4) to be rewritten as

$$y_2 = -\frac{s}{c}y_1 + \left(-\frac{s^2}{c} + c\right)x_2 = -\frac{s}{c}y_1 + \frac{x_2}{c}. \tag{4.6}$$

Note that (4.5) and (4.6) can be rewritten as

$$\begin{bmatrix} x_1 \\ y_2 \end{bmatrix} = \widetilde{G}\begin{bmatrix} y_1 \\ x_2 \end{bmatrix}, \quad \widetilde{G} = \begin{bmatrix} 1/c & s/c \\ -s/c & 1/c \end{bmatrix},$$

and $\widetilde{G}$ is an orthogonal Givens rotation. As multiplication of a vector by a Givens rotation is a stable process, this suggests that the computation of y_2 is likely to be more stable using (4.6) than using (4.4). We call (4.3), (4.6) the *mixed application* of G to a vector x. Similar formulas can be derived for hyperbolic rotations of type 2. Finally we note that the two matrices G and $\widetilde{G}$ are related by the *exchange operator*, $G = \mathrm{exc}(\widetilde{G})$. The exchange operator has a number of interesting mathematical properties; see Higham [14]. In particular, it maps J-orthogonal matrices to orthogonal matrices and vice-versa.

We express the application of unified rotations as follows.

```
function B = r_apply(c, s, J, J̃, B)
% Apply hyperbolic rotation defined by c, s, J, and J̃ to 2 × n matrix B.
γ = J(2,2)/J(1,1), σ̃1 = J̃(1,1)
for j = 1:n
    x = B(1,j)
    B(1,j) = cB(1,j) + γsB(2,j)
    if γ = 1
            B(2,j) = −sx + cB(2,j) % Givens rotation
    elseif σ1 = σ̃1
            B(2,j) = −(s/c)B(1,j) + B(2,j)/c % Rotation of type 1
    else
            B(2,j) = −(c/s)B(1,j) − x/s % Rotation of type 2
    end
end
```

The importance of applying hyperbolic rotations to a vector or a matrix in a mixed way is illustrated in section 6.1.

Unified rotations can be used for reducing $C - \lambda J$ to tridiagonal-diagonal form in a way similar to how Givens rotations are used to tridiagonalize a symmetric matrix (Givens method) [10], [19]. Assume that at the beginning of step j the matrix $C = (c_{ij})$ is tridiagonal as far as its first $j-1$ rows and columns are concerned. At the jth step, we introduce zeros in the matrix C in positions (i,j) and (j,i), $j+2 \le i \le n$ using $n-j-1$ unified rotations. The zeroing operations can be done, for example, in the natural order $j+2, j+3, \ldots, n$ or the reverse order. The element in position (i,j) is annihilated by a unified rotation in the plane (k,i), where k is chosen so that $k < j$, $k \ne i$ and $c_{kj} \ne 0$. The signature matrix is modified each time a hyperbolic rotation of type 2 is applied. The matrix Q which accumulates the product of all the

unified rotations satisfies

$$Q^T C Q = T, \qquad Q^T J Q = \widetilde{J} \in \operatorname{diag}_q^n(\pm 1).$$

The number of rotations required is of order $n^2/2$. The reduction fails if at some stage $\sigma_i |c_{ij}| = \sigma_k |c_{kj}| \neq 0$, where σ_j denotes the jth diagonal elements of J.

For the standard case ($J = I$), the most popular choices for the rotation plane (k, i) are $k = j + 1$ or $k = i - 1$, either choice yielding a perfectly stable reduction. However, when $J \neq I$, the choice of k is crucial for the stability of the reduction. Indeed, using a result of Ostrowski [15, p. 224] one can show that inherent relative errors in a symmetric matrix A can be magnified by as much as $\kappa(Q)^2$ in passing to $Q^T A Q$ for any nonsingular Q [8]. Clearly, $\kappa(G) = 1$ for Givens rotations, but for hyperbolic rotations [4]

$$\kappa(G) = \frac{|c| + |s|}{\big||c| - |s|\big|}, \tag{4.7}$$

which can be arbitrarily large. Hence it is advisable to use as few hyperbolic rotations as possible.

Recall that at stage j of the reduction we need to zero all the elements in rows $j + 2$ up to n of the jth column. First, we perform all possible Givens rotations in planes (ℓ, i) with $\sigma_\ell = \sigma_i$, $j + 1 \leq \ell < i \leq n$. At this point, either the stage is finished or there are two nonzero entries left in positions $(j + 1, j)$ and (i, j) with i such that $j+1 < i \leq n$ and $\sigma_{j+1} = -\sigma_i$. Then a single hyperbolic rotation in the plane $(j+1, i)$ does the final elimination. This strategy has two main advantages. First, it reduces the number of hyperbolic rotations used during the reduction process to at most $n-2$. Secondly, it minimizes the risk of having two hyperbolic rotations acting in the same plane. This tends to reduce the growth of rounding errors and increases the chance that the largest condition number of the individual transformations will be of the same order of magnitude as the condition number of the overall transformation Q. The complete algorithm is summarized as follows.

ALGORITHM 4.1 (tridiagonalization by unified rotations). *Given an $n \times n$ symmetric matrix C and a signature matrix $J \in \operatorname{diag}_q^n(\pm 1)$, the following algorithm overwrites C with the tridiagonal matrix $T = Q^T C Q$ and J with $Q^T J Q \in \operatorname{diag}_q^n(\pm 1)$, Q being the product of unified rotations.*

```
for j = 1: n − 2
    i_h = 0, i = n
    while i > j + 1 or i_h > 0
        if i > j + 1
            Find largest k, j + 1 ≤ k ≤ i, such that J_ii = J_kk.
            rot = [k i]
        else
            rot = [j + 1 i_h], i_h = 0
        end
        if rot(1) = rot(2)
            i_h = rot(1)
        else
            [c, s, J_temp] = u_rotate(C(rot, j), J(rot, rot))
            C(rot, j: n) = r_apply(c, s, J(rot, rot), J_temp, C(rot, j: n))
            C(j: n, rot) = r_apply(c, s, J(rot), J_temp, C(j: n, rot))^T
            C(i, j) = 0; C(j, i) = 0, J(rot, rot) = J_temp
```

```
            end
            i = i - 1
        end
    end
```

The major differences between Algorithm 4.1 and Brebner and Grad's pseudosymmetric Givens algorithm [5] are that in the latter algorithm there is no particular strategy to minimize the number of hyperbolic rotations used and the hyperbolic rotations are applied directly to CJ (instead of as in function r_apply above).

4.2. Reduction by unified Householder reflectors. Unified Householder reflectors [4] include standard orthogonal Householder transformations [11] together with hyperbolic Householder reflectors [20], [21]. Given a signature matrix $J = \mathrm{diag}(\sigma_i)$, a unified Householder matrix has the form

$$H = H(J,k,v) = P\Big(J - \frac{2vv^T}{v^TJv}\Big), \quad v^TJv \neq 0, \tag{4.8}$$

where P is a permutation matrix in the $(1,k)$-plane.

For any vector x such that $x^TJx \neq 0$, the unified Householder vector v can be chosen so that H maps x onto the first column of the identity matrix. Let k be such that $e_k^TJe_k = \sigma_k := \mathrm{sign}(x^TJx)$ and let

$$v = Jx + \sigma_k \mathrm{sign}(x_k)|x^TJx|^{1/2}e_k. \tag{4.9}$$

Then it is easy to check that $v^TJv \neq 0$ and that $Hx = -\sigma_k\mathrm{sign}(x_k)|x^TJx|^{1/2}e_1$. Note also that P^TH is J-orthogonal.

The application of a hyperbolic Householder matrix to a vector can be done either directly, as

$$Hx = P\left(Jx - \frac{2v^Tx}{v^TJv}v\right),$$

or, as for hyperbolic rotations, in a mixed way making use of the orthogonal matrix $\mathrm{exc}(H)$. Stewart and Stewart [23] show that both approaches are mixed-forward backward stable. We use the first approach since it yields simpler coding.

In [4] it is shown that

$$\sigma_{\min}^{-1}(H) = \sigma_{\max}(H) = \frac{v^Tv}{|v^TJv|} + \sqrt{\left(\frac{v^Tv}{v^TJv}\right)^2 - 1}. \tag{4.10}$$

For $J = I$, $\sigma_{\min} = \sigma_{\max} = 1$ and for $J \neq I$ the ratio v^Tv/v^TJv can be arbitrarily large. Fortunately, there is some freedom in the choice of the plane $(1,k)$ for the permutation P. Choosing k so that

$$e_k^TJe_k = \mathrm{sign}(x^TJx) \quad \text{and} \quad |x_k| \text{ is maximized}$$

minimizes the ratio v^Tv/v^TJv and therefore minimizes $\kappa(H)$. This is the pivoting strategy proposed in [4], [23].

The following pseudocode inspired by [4, Alg. 3] determines the permutation matrix P and constructs the unified Householder vector.

```
function [v, k, β, α] = u_house(x, J)
% Determine the permutation P in the (1, k) plane and compute v, α, and β
```

% such that $H = P(J - \beta vv^T)$ satisfies $Hx = -\alpha e_1$ with $P^T H$ J-orthogonal.
if $x^T Jx = 0$
    No hyperbolic Householder exists—abort.
end
$m = \|x\|_\infty$, $x = x/m$
if $J = \pm I$
    $k = 1$
else
    Find k so that $|x_k|$ is maximized and $\text{sign}(x_k^T Jx_k) = J_{kk}$.
end
$\alpha = J_{kk}\text{sign}(x_k)|x^T Jx|^{1/2}$
$v = Jx + \alpha e_k$
$\beta = 2/(v^T Jv)$
$\alpha = m\alpha$

The symmetric matrix C can be reduced to tridiagonal form while keeping the diagonal form of J by $n-2$ unified Householder transformations. Each transformation annihilates the required part of a whole column and whole corresponding row. The complete algorithm is summarized below.

ALGORITHM 4.2 (tridiagonalization by unified Householder reflectors). *Given an $n \times n$ symmetric matrix C and a signature matrix $J \in \text{diag}_q^n(\pm 1)$, the following algorithm overwrites C with the tridiagonal matrix $T = Q^T CQ$ and J with $Q^T JQ \in \text{diag}_q^n(\pm 1)$, Q being the product of unified Householder reflectors.*

for $j = 1\colon n-2$
    $ind = j+1\colon n$
    $[v, k, \beta, \alpha] = \text{u\_house}(C(ind, j), J(ind, ind))$
    Swap rows and columns $j+1$ and $j+k$ of C.
    $C(ind, j) = -\alpha e_1$, $C(j, ind) = C(ind, j)^T$
    $p = \beta J(ind, ind)C(ind, ind)v$
    $w = p - \beta^2(v^T C(ind, ind)v)v/2$
    $C(ind, ind) = J(ind, ind)C(ind, ind)J(ind, ind) - wv^T - vw^T$
end

Note that the reduction fails if at some stage j, $x^T Jx = 0$, where $x = C(j+1\colon n, j)$.

Algorithm 4.2 differs from the pseudosymmetric Householder algorithm in [5] in that, for the latter algorithm, Brebner and Grad use a rank-one update H of the form $H = I - 2Jvv^T$, where v can have complex entries even though H is real. This vector is not computed but, instead, the transformation H is computed element by element and applied explicitly to CJ, which is a costly operation. Also, no pivoting is used to reduce the condition number of the transformations.

4.3. Reduction by a mix of Householder reflectors and hyperbolic rotations. Here we adapt an idea developed by Bojanczyk, Higham, and Patel [3] for hyperbolic QR factorizations of rectangular matrices. We propose a tridiagonalization that uses a combination of Householder reflectors and hyperbolic rotations. As hyperbolic rotations are not norm-preserving, we aim to use a minimal number of them.

Assume for notational simplicity that $J = \text{diag}(I_p, -I_q) \in \text{diag}_q^n(\pm 1)$ and partition $x \in \mathbb{R}^n$ so that $x_p = x(1\colon p)$ and $x_q = x(p+1\colon n)$. We first define a $(J, \widetilde{J})$-orthogonal matrix that maps x into the first column of the identity matrix. Let H_p and H_q be two Householder matrices defined so that

$$H_p x_p = -\|x_p\| e_1, \qquad H_q x_q = -\|x_q\| e_1.$$

Next we define a 2×2 hyperbolic rotation such that

$$\begin{bmatrix} c & -s \\ -s & c \end{bmatrix} \begin{bmatrix} \|x_p\| \\ \|x_q\| \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}, \quad \alpha \in \mathbb{R},$$

and build from it an $n \times n$ hyperbolic rotation G in the $(1, p+1)$ plane. Then the matrix

$$S = G \begin{bmatrix} H_p & 0 \\ 0 & H_q \end{bmatrix} \tag{4.11}$$

maps x into the first column of the identity matrix and satisfies $\widetilde{J} \equiv S^T J S \in \operatorname{diag}_q^n(\pm 1)$. Note that $\widetilde{J} = J$ when G is a hyperbolic rotation of type 1 and if G is a hyperbolic rotation of type 2 then $\widetilde{J}$ and J are identical except in position $(1,1)$ and $(p+1, p+1)$, where their signs differ. From (4.11) and (4.7), the condition number of S is given by

$$\kappa(S) = \frac{\|x_p\| + \|x_q\|}{\big|\|x_p\| - \|x_q\|\big|}. \tag{4.12}$$

Unlike for the tridiagonalization via unified Householder matrices, we have no free parameters that can be used to minimize $\kappa(S)$. The next result shows that $\kappa(S)$ is already of optimal condition relative to unified Householder matrices.

THEOREM 4.3. *Let H be a unified Householder reflector as in* (4.8) *and let S be a combination of Householder reflectors and a hyperbolic rotation as in* (4.11)*, both mapping a vector x to a multiple of e_1, the first column of the identity matrix. Then*

$$\kappa(S) \le \kappa(H).$$

If R denotes the matrix which accumulates the product of the Givens rotations and the hyperbolic rotation mapping x to a multiple of e_1 as described in section 4.1*, then*

$$\kappa(R) = \kappa(S).$$

Proof. Let $H = P(J - \beta v v^T)$, where

$$\beta = 2/v^T J v, \quad v = Jx + \sigma_k \operatorname{sign}(x_k) |x^T J x|^{1/2} e_k \tag{4.13}$$

for some k such that $e_k^T J e_k = \sigma_k = \operatorname{sign}(x^T J x)$ and P is a permutation in the $(1,k)$-plane. Assume that $J = (I_p, -I_q)$ and partition v and x accordingly:

$$\begin{aligned} v_p &= v(1\colon p), & x_p &= x(1\colon p), \\ v_q &= v(p+1\colon p+q), & x_q &= x(p+1\colon p+q). \end{aligned}$$

From (4.10),

$$\kappa(H) = \frac{\sigma_{\max}(H)}{\sigma_{\min}(H)} = \left(\frac{v^T v + \sqrt{(v^T v)^2 - (v^T J v)^2}}{v^T J v} \right)^2 = \left(\frac{\|v_p\| + \|v_q\|}{\|v_p\| - \|v_q\|} \right)^2.$$

Suppose that $\kappa(H) < \kappa(S)$. There is no loss of generality in assuming that $\|x_p\| > \|x_q\|$. Using the expression for $\kappa(S)$ in (4.12) we have

$$(\|x_p\| + \|x_q\|)(\|v_p\| - \|v_q\|)^2 > (\|x_p\| - \|x_q\|)(\|v_p\| + \|v_q\|)^2,$$

or equivalently,

$$2\|x_p\|\|v_p\|\|v_q\| - \|x_q\|\|v_p\|^2 - \|x_q\|\|v_q\|^2 < 0. \tag{4.14}$$

Since $\|x_p\| > \|x_q\|$, $\operatorname{sign}(x^T Jx) > 0$ and hence $1 \le k < p$. From (4.13), $\|v_q\| = \|x_q\|$ and

$$\|v_p\|^2 = 2\|x_p\|^2 - \|x_q\|^2 + 2|x_k|(\|x_p\|^2 - \|x_q\|^2)^{1/2} = \|x_p\|^2(2\mu - \alpha),$$

where $\mu = 1 + |x_k|(1-\alpha)^{1/2}/\|x_p\|$ and $\alpha = \|x_q\|^2/\|x_p\|^2$. Using these expressions for $\|v_p\|$ and $\|v_q\|$ in inequality (4.14) leads to

$$f(\mu) = 3\mu^2 - 4\mu\alpha + \alpha^2 < 0,$$

which is satisfied for $\frac{\alpha}{3} < \mu < \alpha < 1$. But by the definition of μ, $1 \le \mu \le 1+(1-\alpha)^{1/2}$ and for these values of μ, $f(\mu) \ge 0$. Hence $\kappa(H) \ge \kappa(S)$.

The equality $\kappa(R) = \kappa(S)$ is obvious. □

Assume that (C, J) has been permuted so that $J = \operatorname{diag}(I_p, -I_q)$. Again, as in the previous section, we can transform C to tridiagonal form while preserving the diagonal form of J by $n-2$ transformations of the form (4.11). The key point in the reduction is that at each step the part of the signature matrix involved in the transformation is of the form $\operatorname{diag}(I_{\tilde{p}_j}, -I_{\tilde{q}_j})$, $\tilde{p}_j + \tilde{q}_j = n - j$. Note that if we reach the stage where $\tilde{p}_j = 0$ or $\tilde{q}_j = 0$ then the rest of the reduction is carried out with orthogonal Householder matrices only.

This reduction uses at least $\min(p-1, q)$ hyperbolic rotations and at most $n-2$. The smallest number $\min(p-1, q)$ occurs when all the transformations in the reduction process are derived from hyperbolic rotations of type 1. The largest number, $n-2$, happens if hyperbolic rotations of type 2 are used at each step of the reduction. Note that the reduction fails if at some stage j, $\|x_{\tilde{p}_j}\| = \|x_{\tilde{q}_j}\| \ne 0$.

ALGORITHM 4.4 (tridiagonalization by mixed Householder reflectors-hyperbolic rotations). *Given an $n \times n$ symmetric matrix C and a signature matrix $J = (I_p, -I_q)$, the following algorithm overwrites C with the tridiagonal matrix $T = Q^T CQ$ and J with $Q^T JQ \in \operatorname{diag}_q^n(\pm 1)$, Q being the product of mixed Householder reflectors–hyperbolic rotations.*

```
for j = 1:n-2
    p = max(0, p-1)
    if p > 1
        ind = j+1:j+p
        [v, k, β, α] = u_house(C(j+1:j+p, j), J(ind, ind))
        C(ind, j:n) = C(ind, j:n) − βv(v^T C(ind, j:n))
        C(j:n, ind) = C(j:n, ind) − β(C(j:n, ind)v)v^T
    end
    if q > 1
        ind = (j+p+1:n)
        [v, k, β, α] = u_house(C(ind, j), J(ind, ind))
        C(ind, j:n) = C(ind, j:n) − βv(v^T C(ind, j:n))
        C(j:n, ind) = C(j:n, ind) − β(C(j:n, ind)v)v^T
    end
    if p > 0 and q > 0
        rot = [j+1, j+p+1]
        [c, s, J_temp] = u_rotate(C(rot, j), J(rot, rot))
```

```
            C(rot, j:n) = r_apply(c, s, J(rot, rot), J_temp, C(rot, j:n))
            C(j:n, rot) = r_apply(c, s, J(rot), J_temp, C(j:n, rot)^T))^T
            C(i, j) = 0; C(j, i) = 0
            if J(rot, rot) = -J_temp
                  p = p + 1, q = q - 1
                  J(rot, rot) = J_temp
            end
        end
    end
```

We cannot conclude from Theorem 4.3 that Algorithms 4.1 and 4.4 are more stable than Algorithm 4.2 since at step k of the tridiagonalization process the column of C to be annihilated is not the same for each reduction. However, intuitively, we may expect Algorithms 4.1 and 4.4 to behave better than Algorithm 4.2.

5. Monitoring condition numbers and preventing breakdown. If serious breakdown occurs during the reduction to tridiagonal-diagonal form (see the end of section 2) then we can permute C and start again. This is equivalent to restarting the Lanczos process described in the proof of Theorem 2.1 with a new vector q_1. Of course, the major disadvantage with this approach is that all the previous computation is lost. We take an alternative approach, based on an idea from Geist, Lu, and Wachpress [9] for curing breakdown occurring in the tridiagonalization of nonsymmetric matrices.

If breakdown occurs at step j of the reduction process or if the condition number of the next transformation is too large, we apply a unified rotation $\widetilde{G}$ on the two first rows and columns of the current C. This brings nonzero values in positions $(3,1)$ and $(1,3)$. This bulge in the tridiagonal form is chased down the matrix from position $(3,1)$ to $(4,2)$ and so on via $j-2$ unified rotations. This chasing procedure costs $O(j)$ operations and the result is a new column j in C. The whole procedure may be tried again if some large condition numbers occurs before the reduction is completed.

In our implementation the unified rotation $\widetilde{G}$ is generated randomly but with the constraint that $\kappa(\widetilde{G}) = O(1)$.

6. Numerical experiments. Our aim in this section is to investigate the numerical properties of the tridiagonal-diagonal reduction algorithms just described. We name our MATLAB implementations

- `trd_ur`: tridiagonalization by unified rotations (Algorithm 4.1),
- `trd_uh`: tridiagonalization by unified Householder reflectors (Algorithm 4.2),
- `trd_hr`: tridiagonalization by mixed Householder reflectors-hyperbolic rotations (Algorithm 4.4).

Given a symmetric matrix C and a signature matrix J we formed explicitly, during the course of the reduction, the transformation Q such that $T = Q^T C Q$ is tridiagonal and $\widetilde{J} = Q^T J Q$ is a signature matrix. The following quantities were computed:

- the scaled residual error and departure from $(J, \widetilde{J})$-orthogonality

$$\mathcal{R} = \frac{\|Q^T C Q - T\|}{\|C\|\|Q\|^2}, \qquad \mathcal{O} = \frac{\|Q^T J Q - \widetilde{J}\|}{\|Q\|^2}, \tag{6.1}$$

- $\kappa(Q)$, the condition number of the transformation Q,
- the largest condition numbers,

$$\kappa_G = \max_k \kappa(G_k), \quad \kappa_H = \max_k \kappa(H_k), \quad \kappa_S = \max_k \kappa(S_k),$$

of the transformations used to zero parts of the matrix C. Here G, H, and S

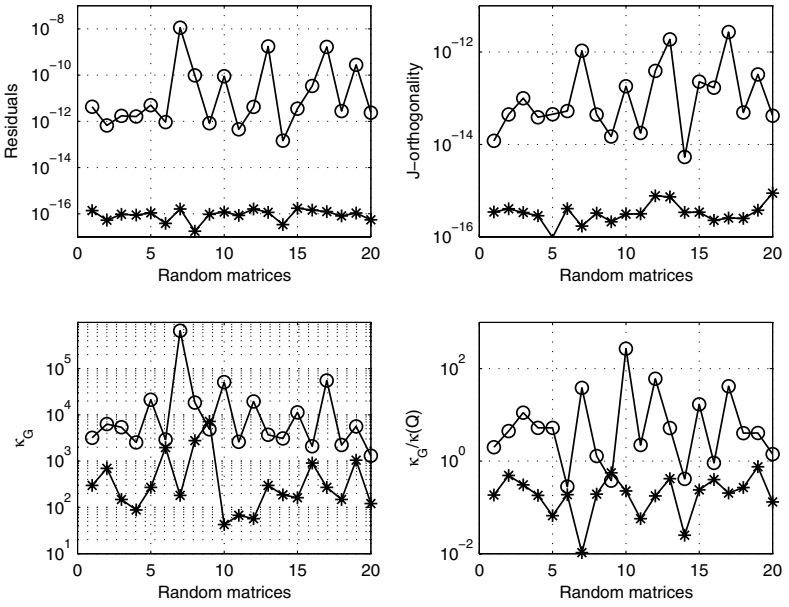


FIG. 6.1. *Residuals and condition numbers for* 20 *random matrices. Results from* `trd_BG1` *are marked with* "o" *and results from* `trd_ur` *are marked with* "$*$."

refer to unified rotation, unified Householder reflector, and a combination of two Householder reflectors and one hyperbolic rotation, respectively.

6.1. Tridiagonalization by unified rotations. We first compare `trd_ur` to an implementation of Brebner and Grad's pseudosymmetric Givens method named `trd_BG1`. We ran a set of tests with matrices of the form

```
C = randn(n); C = C+C'; J = mysign(randn(n));
```

where `mysign` is a sign function defined so that `mysign`$(0) = 1$. The residual $\mathcal{R}$ and the departure from $(J, \widetilde{J})$-orthogonality $\mathcal{O}$ as defined in (6.1) are plotted on the top left and right in Figure 6.1 for twenty random matrices of size $n = 50$. Results obtained by `trd_BG1` are plotted with "o" and we use "$*$" for results from `trd_ur`. On this set of matrices, the residuals $\mathcal{R}$ and $\mathcal{O}$ from `trd_ur` are smaller than the ones from `trd_BG1` by a factor as large as 10^7 for $\mathcal{R}$ and 10^4 for $\mathcal{O}$. For a given test problem (C, J), `trd_BG1` and `trd_ur` both compute the same Q, but the construction of Q differs since it is obtained by a different sequence of transformations. The left-hand plot at the bottom of Figure 6.1 helps to compare the largest condition numbers κ_G of the individual transformations used by each algorithm during the reduction process. It shows that κ_G is nearly always smaller for `trd_ur`. Not surprisingly, large values of κ_G correspond to test problems with large values of $\mathcal{R}$ and $\mathcal{O}$. The right-hand plot at the bottom of Figure 6.1 compares both algorithms' ratios $\kappa_G/\kappa(Q)$. Interestingly, for `trd_ur`, κ_G is always smaller than the condition number of the overall transformation Q whereas κ_G is in general larger than $\kappa(Q)$ for `trd_BG1`. The four plots on Figure 6.1 illustrate the numerical superiority of our tridiagonalization using unified rotations over Brebner and Grad's pseudosymmetric Givens method. The improvements are due to the way we apply the rotations and our zeroing strategy.

TABLE 6.1
Comparison between explicit and implicit application of hyperbolic rotations to matrices.

$\mathcal{R}_d$	$\mathcal{R}_m$	$\kappa(Q)$	κ_G	$\mathcal{E}_d$	$\mathcal{E}_m$	$\mathrm{cond}(\lambda)$
2×10^{-12}	2×10^{-15}	3.02	2×10^{3}	4×10^{-10}	2×10^{-13}	4×10^{2}

To emphasize the fact that how hyperbolic rotations are applied to a matrix may be crucial to the stability of the computation we use the direct search maximization routine `mdsmax` of the MATLAB Matrix Computation Toolbox [12] to maximize both ratios $\mathcal{R}_d/\mathcal{R}_m$ and $\mathcal{R}_m/\mathcal{R}_d$. The subscripts d and m stand for direct and mixed, respectively, depending on how the hyperbolic rotations are applied to C during the course of the reduction. We used `trd_BG1` with an option on how to apply the rotations. We found that for some matrix pairs (C, J), $\mathcal{R}_d \gg \mathcal{R}_m$ but when $\mathcal{R}_m$ is larger than $\mathcal{R}_d$, $\mathcal{R}_m \lesssim \mathcal{R}_d$ always. Table 6.1 provides some relevant quantities for a 5×5 pair (C, J) generated by `mdsmax`. We also compared the eigenvalues λ_i of the initial pair (C, J) with those $\tilde{\lambda}_i$ of $(T, \widetilde{J})$ and their corresponding relative condition numbers $\mathrm{cond}(\lambda_i)$,

$$\mathrm{cond}(\lambda_i) = \frac{\|x_i\|\|y_i\|}{|\lambda_i|\ |y_i^* J x_i|},$$

where x_i and y_i are the corresponding right and left eigenvectors. We denote by

$$\mathcal{E} = \max_{i=1:n} \frac{|\lambda_i - \tilde{\lambda}_i|}{|\lambda_i|}$$

the largest relative error for the computed eigenvalues. For this particular example, $\mathcal{R}_d \approx 10^3 \mathcal{R}_m$. Since $\kappa(Q) = O(1)$, it is reasonable to expect $\mathcal{R} = O(u)$ which is clearly not the case when direct application of unified rotations is used. The table also shows that a large value for the residual $\mathcal{R}_d$ directly affects the accuracy to which the eigenvalues are computed from $(T, \widetilde{J})$.

6.2. Tridiagonalization by unified Householder reflectors. We now compare `trd_uh` to an implementation of Brebner and Grad's pseudosymmetric Householder method named `trd_BG2`. The main numerical difference between the two algorithms is that `trd_uh` uses a pivoting strategy aimed to reduce the condition numbers of the unified Householder reflectors. We ran a sequence of tests similar to the ones described in section 6.1. Results are plotted in Figure 6.2 for twenty random test problems of dimension 50. These plots clearly illustrate that the pivoting strategy helps to reduce the residuals and the departure from $(J, \widetilde{J})$-orthogonality. For this set of examples, $\mathcal{R}$ and $\mathcal{O}$ are reduced on average by a factor 10^2 and 10, respectively; the reduction factor is as large as 10^3 for $\mathcal{R}$ and as large as 10^2 for $\mathcal{O}$. As expected, κ_H for `trd_uh` is always smaller than κ_H for `trd_BG2` by a factor as large as 10^3. Recall that small κ_H are essential for the stability of the reduction.

6.3. Comparison of the three reductions. For a particular symmetric-diagonal pair (C, J) with $J = \mathrm{diag}(I_p, -I_q)$ we know that, from Theorem 2.1, the three algorithms produce up to signs the same matrix Q and tridiagonal matrix T. They differ numerically in the way Q is formed.

We generated a large set of problems with matrices C of the form

```
C = randn(n); C = C+C'
```

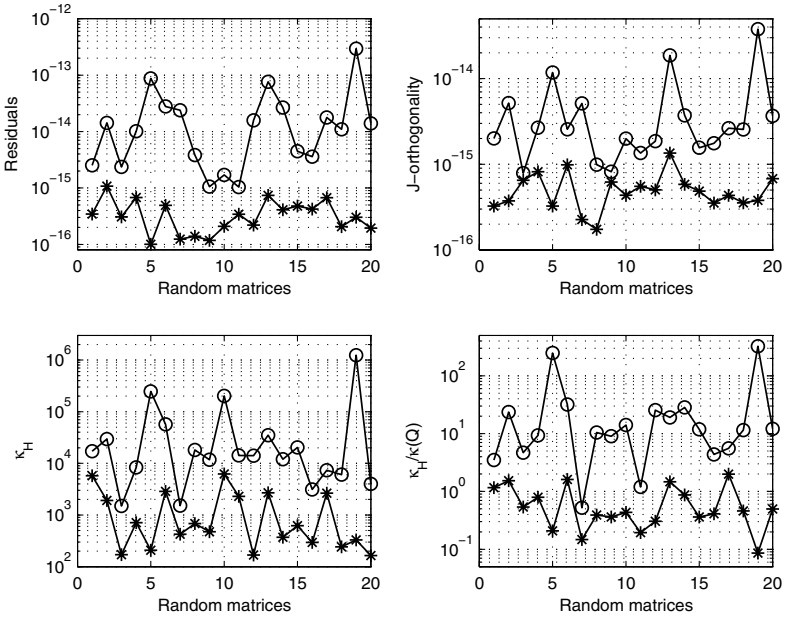


FIG. 6.2. *Residuals and condition numbers for* 20 *random matrices. Results from* `trd_BG2` *are marked with* "o" *and results from* `trd_uh` *are marked with* "$*$."

and

```
C = gallery('randsvd',n); C = C+C'
```

and also matrices $C = Q^T T Q$ obtained from random tridiagonal matrices T and random J-orthogonal matrices Q with prescribed condition numbers. Higham's algorithm [14] was used to generate the random Q.

We ran extensive tests with these types of problems. Here is a summary of our findings.

- As expected, `trd_ur`, `trd_uh` yield residuals of the same order of magnitude.
- 80% of the time, `trd_uh` has residuals of the same order of magnitude as `trd_hr` or `trd_ur`.
- In 20% of the cases where the residuals have different orders of magnitude, `trd_uh` appears the least stable. On average, the residuals and departure from $(J, \widetilde{J})$-orthogonality are 10 times larger with `trd_uh` than with `trd_ur` or `trd_hr`.
- Most of the time, κ_G and κ_S are smaller than κ_H, which is consistent with the previous bullet. Large condition numbers for the individual transformations directly affect the residuals.
- When $\kappa(Q)$ is large the $(J, \widetilde{J})$-departure from orthogonality of Q tends to be larger with `trd_uh` than with the two others algorithms.

This battery of tests seems to indicate that amongst the three reductions `trd_uh` is the least stable. Since `trd_ur` is nearly twice more costly than `trd_hr`, we suggest to use the latter, that is, to use a combination of Householder reflectors and hyperbolic rotations (Algorithm 4.4) to reduce a symmetric-diagonal pair to tridiagonal-diagonal form. We would like to emphasize that in most instances the three algorithms all produce residuals close to what we would expect from a stable algorithm.

Acknowledgment. I thank the referees for valuable suggestions that improved the paper.

REFERENCES

[1] D. A. Bini, L. Gemignani, and F. Tisseur, *The Ehrlich-Aberth Method for the Nonsymmetric Tridiagonal Eigenvalue Problem*, Numerical Analysis Report 428, Manchester Centre for Computational Mathematics, Manchester, England, 2003.

[2] A. Bojanczyk, R. P. Brent, P. Van Dooren, and F. R. de Hoog, *A note on downdating the Cholesky factorization*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 210–221.

[3] A. Bojanczyk, N. J. Higham, and H. Patel, *Solving the indefinite least squares problem by hyperbolic QR factorization*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 914–931.

[4] A. W. Bojanczyk, S. Qiao, and A. O. Steinhardt, *Unifying unitary and hyperbolic transformations*, Linear Algebra Appl., 316 (2000), pp. 183–197.

[5] 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.

[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] P. I. Davies, N. J. Higham, and F. Tisseur, *Analysis of the Cholesky method with iterative refinement for solving the symmetric definite generalized eigenproblem*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 472–493.

[8] S. D. Garvey, F. Tisseur, M. I. Friswell, and J. E. T. Penny, *Simultaneous tridiagonalization of two symmetric matrices*, Int. J. Numer. Meth. Engng., (2003), pp. 1643–1660.

[9] G. A. Geist, A. Lu, and E. L. Wachpress, *Stabilized Gaussian Reduction of an Arbitrary Matrix to Tridiagonal Form*, Tech. report, Report ORNL/TM-11089, Oak Ridge National Laboratory, TN, 1989.

[10] W. J. Givens, *Numerical Computation of the Characteristic Values of a Real Symmetric Matrix*, Tech. Report ORNL-1574, Oak Ridge National Laboratory, Oak Ridge, TN, 1954.

[11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[12] N. J. Higham, *The Matrix Computation Toolbox*, http://www.ma.man.ac.uk/~higham/mctoolbox.

[13] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, 2002.

[14] N. J. Higham, *J-orthogonal matrices: Properties and generation*, SIAM Rev., 45 (2003), pp. 504–519.

[15] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[16] Z. S. Liu, *On the Extended HR Algorithm*, Technical Report PAM-564, Center for Pure and Applied Mathematics, University of California, Berkeley, CA, 1992.

[17] R. Onn, A. O. Steinhardt, and A. W. Bojanczyk, *The hyperbolic singular value decomposition and applications*, IEEE Trans. Signal Processing, 39 (1991), pp. 1575–1588.

[18] B. N. Parlett, *Reduction to tridiagonal form and minimal realizations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 567–593.

[19] B. N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, 1998. Corrected reprint of the 1980 original.

[20] C. M. Rader and A. O. Steinhardt, *Hyperbolic Householder transformations*, IEEE Trans. Acoust. Speech Signal Processing, ASSP-34 (1986), pp. 1589–1602.

[21] C. M. Rader and A. O. Steinhardt, *Hyperbolic Householder transforms*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 269–290.

[22] I. Slapničar, *Componentwise analysis of direct factorization of real symmetric and Hermitian matrices*, Linear Algebra Appl., 272 (1998), pp. 227–275.

[23] M. Stewart and G. W. Stewart, *On hyperbolic triangularization: Stability and pivoting*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 847–860.

[24] F. Uhlig, *The $\mathcal{DQR}$ algorithm, basic theory, convergence, and conditional stability*, Numer. Math., 76 (1997), pp. 515–553.

[25] D. 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.

[26] R. Zurmühl and S. Falk, *Matrizen und ihre Anwendungen für angewandte Mathematiker, Physiker und Ingenieure. Teil* 2, 5th ed. Springer-Verlag, Berlin, 1986.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 1, pp. 233–251

CONVERGENCE OF GMRES FOR TRIDIAGONAL TOEPLITZ MATRICES*

J. LIESEN† AND Z. STRAKOŠ‡

Abstract. We analyze the residuals of GMRES [Y. Saad and M. H. Schultz, *SIAM J. Sci. Statist. Comput.*, 7 (1986), pp. 856–859], when the method is applied to tridiagonal Toeplitz matrices. We first derive formulas for the residuals as well as their norms when GMRES is applied to scaled Jordan blocks. This problem has been studied previously by Ipsen [*BIT*, 40 (2000), pp. 524–535] and Eiermann and Ernst [*Private communication,* 2002], but we formulate and prove our results in a different way. We then extend the (lower) bidiagonal Jordan blocks to tridiagonal Toeplitz matrices and study extensions of our bidiagonal analysis to the tridiagonal case. Intuitively, when a scaled Jordan block is extended to a tridiagonal Toeplitz matrix by a superdiagonal of small modulus (compared to the modulus of the subdiagonal), the GMRES residual norms for both matrices and the same initial residual should be close to each other. We confirm and quantify this intuitive statement. We also demonstrate principal difficulties of any GMRES convergence analysis which is based on eigenvector expansion of the initial residual when the eigenvector matrix is ill-conditioned. Such analyses are complicated by a cancellation of possibly huge components due to close eigenvectors, which can prevent achieving well-justified conclusions.

Key words. Krylov subspace methods, GMRES, minimal residual methods, convergence analysis, Jordan blocks, Toeplitz matrices

AMS subject classifications. 15A09, 65F10, 65F20

DOI. 10.1137/S0895479803424967

1. Introduction.

Consider solving a linear algebraic system $Ax = b$, real or complex, where A is an N by N nonsingular matrix with GMRES [9]. Starting from an initial guess x_0, this method computes the initial residual $r_0 = b - Ax_0$ and a sequence of iterates, $x_1, x_2, \ldots$ so that the nth residual $r_n = b - Ax_n$ satisfies

$$\|r_n\| \;=\; \|p_n(A)r_0\| \;=\; \min_{p\in\pi_n} \|p(A)r_0\|, \tag{1.1}$$

where π_n denotes the set of polynomials of degree at most n with value one at the origin and $\|\cdot\|$ denotes the 2-norm. It is easy to see from (1.1) that (in exact arithmetic) the GMRES algorithm terminates, i.e., computes the solution x, in at most N steps. We also wish to point out that, unless there is a well-justified reason for choosing a nonzero initial approximation, one should consider $x_0 = 0$ (see [8]).

Suppose that the vectors $r_0, Ar_0, \ldots, A^n r_0$ generating the $(n+1)$st Krylov subspace $\mathcal{K}_{n+1}(A, r_0) = \mathrm{span}\{r_0, Ar_0, \ldots, A^n r_0\}$ are linearly independent. Then r_n is a nonzero vector and GMRES cannot terminate before the step $n+1$. Denote by K_{n+1} the matrix of the Krylov vectors,

$$K_{n+1} \;=\; [r_0, Ar_0, \ldots, A^n r_0] \;\equiv\; [r_0, W_n R_n]\,, \tag{1.2}$$

*Received by the editors March 17, 2003; accepted for publication (in revised form) by L. Elden January 9, 2004; published electronically September 14, 2004.
http://www.siam.org/journals/simax/26-1/42496.html

†Institute of Mathematics, Technical University of Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany (liesen@math.tu-berlin.de). The work of this author was supported by the Emmy Noether-Programm of the Deutsche Forschungsgemeinschaft.

‡Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vod. věží 2, 182 07 Prague, and Technical University Liberec, Hálkova 6, 461 17 Liberec, Czech Republic (strakos@cs.cas.cz, http://www.cs.cas.cz/~strakos). The work of this author was supported by the GA CR under grant 201/02/0595 and by the Ministry of Education of the Czech Republic under project MSM242200002.

where W_n has orthonormal columns and R_n is upper triangular.

In [5, Theorem 2.1] Ipsen shows that r_n is determined by the first row of the Moore–Penrose pseudoinverse of K_{n+1},

$$r_n^T \;=\; \|r_n\|^2 \, e_1^T K_{n+1}^+. \tag{1.3}$$

Based on this result she argues that as long as the matrix K_{n+1} is well-conditioned, the decrease of the GMRES residual norms in the steps 1 to n must be slow. Then she applies this relation to analyze the GMRES behavior for scaled Jordan blocks [5, Theorem 3.1].

In [6, pp. 1505–1506], it is shown that

$$r_n^T \;=\; \|r_n\|^2 \, e_1^T [r_0, W_n]^+ \,, \tag{1.4}$$

which refines Ipsen's argument about the relation between ill-conditioning of the Krylov matrix and convergence of the GMRES residual norms. The proofs in [6] are based on the elementary geometrical interpretation of the pseudoinverse (orthogonality relations).

In this paper we study the GMRES residuals for linear systems with tridiagonal Toeplitz matrices T. We start with results analogous to those of Ipsen for scaled Jordan blocks, and then we analyze their extensions. We are particularly interested in the case when the entries on the superdiagonal of T are significantly smaller in modulus (absolute value) than the entries on the subdiagonal. This represents an example of very large eigenvector conditioning (even infinite when the matrix reduces to a scaled Jordan block); i.e., we deal with highly nonnormal matrices. Rather than applying a worst-case analysis based on properties of the matrix T only, we exploit the structure of T and relate the GMRES convergence to the structure and numerical values of the entries of the initial residual r_0. This allows qualitative as well as quantitative statements about the influence of T *as well as* r_0 on the GMRES residuals. In proofs, we follow, as in [6, pp. 1505–1506], the elementary orthogonality idea.

Analytic results for scaled Jordan blocks and general tridiagonal Toeplitz matrices are given in sections 2 and 3, respectively. Section 4 shows numerical experiments, and section 5 contains concluding remarks. In this paper we do not consider rounding errors, i.e., we assume exact arithmetic.

2. Scaled Jordan blocks. For given nonzero parameters γ and λ, consider an N by N scaled Jordan block J,

$$J \;=\; \gamma S + \lambda I \;\equiv\; \gamma(S + \tau I)\,, \qquad \tau \;\equiv\; \frac{\lambda}{\gamma}\,, \tag{2.1}$$

where I is the identity and $S = [e_2, \ldots, e_N, 0]$ is the down shift matrix (e_j denotes the jth vector of the standard Euclidean basis). The scaling does not affect GMRES convergence; it is used for convenience only. The GMRES residual norms for systems with scaled Jordan blocks have been studied in [2] and in [5, section 3]. Here we study the same problem, but we formulate and prove our results differently from [2, 5].

THEOREM 2.1. *Suppose that GMRES is applied to a system with the matrix* $J = \gamma(S + \tau I)$ *and the initial residual* $r_0 = [\rho_1, \ldots, \rho_N]^T$. *Let* ρ_l *be the first nonzero entry of* r_0. *Then for* $n = 0, 1, \ldots, N - l$ *the GMRES residuals satisfy*

$$r_n^T = \|r_n\|^2 \, [1, -\tau, \ldots, (-\tau)^n] \, [r_0, Sr_0, \ldots, S^n r_0]^+ \,, \tag{2.2}$$

$$\|r_n\| \geq \left(\sum_{j=0}^{n} |\tau|^{2j}\right)^{-\frac{1}{2}} \sigma_{\min}\left([r_0, Sr_0, \ldots, S^n r_0]\right), \tag{2.3}$$

and $r_{N-l+1} = 0$*, where* $\sigma_{\min}(X)$ *denotes the minimal singular value of the matrix* X*. Furthermore, for* $n = 0, 1, \ldots, N-l$*,*

$$\|r_n\| \;\leq\; (n+1)^{\frac{1}{2}} \|r_0\| \left(\sum_{j=0}^{n} |\tau|^{2j}\right)^{-\frac{1}{2}}. \tag{2.4}$$

Proof. Since $\mathcal{K}_{n+1}(J, r_0) = \mathcal{K}_{n+1}(S, r_0)$ and $\rho_l \neq 0$, it is easy to see that for $n = 0, 1, \ldots, N-l$ the matrices $[r_0, Jr_0, \ldots, J^n r_0]$ have full column rank. Hence, for $n = 0, 1, \ldots, N-l$, (1.3) (see also [6, Theorem 2.1]) shows that

$$r_n^T \;=\; \|r_n\|^2 \, e_1^T [r_0, Jr_0, \ldots, J^n r_0]^+ \;\equiv\; \|r_n\|^2 \, g_n^T. \tag{2.5}$$

The identity $[r_0, Jr_0, \ldots, J^n r_0]^+ [r_0, Jr_0, \ldots, J^n r_0] \;=\; I$ gives

$$g_n^T [r_0, Jr_0, \ldots, J^n r_0] \;=\; e_1^T.$$

We next prove, by induction,

$$g_n^T [r_0, Sr_0, \ldots, S^n r_0] \;=\; [1, -\tau, \ldots, (-\tau)^n]. \tag{2.6}$$

Clearly,

$$0 = g_n^T J r_0 = \gamma \, g_n^T S r_0 + \lambda \, g_n^T r_0 = \gamma \, g_n^T S r_0 + \lambda, \qquad \text{i.e., } g_n^T S r_0 = -\tau,$$

and the general step,

$$\begin{aligned}
0 = g_n^T J^k r_0 &= g_n^T (\gamma S + \lambda I)^k r_0 \\
&= g_n^T \left(\sum_{j=0}^{k} \begin{pmatrix} k \\ j \end{pmatrix} \gamma^{k-j} \lambda^j S^{k-j} \right) r_0 \\
&= \gamma^k g_n^T S^k r_0 + \sum_{j=1}^{k} \begin{pmatrix} k \\ j \end{pmatrix} \gamma^{k-j} \lambda^j (-\tau)^{k-j} \\
&= \gamma^k g_n^T S^k r_0 - (-\lambda)^k + \sum_{j=0}^{k} \begin{pmatrix} k \\ j \end{pmatrix} (-\lambda)^{k-j} \lambda^j \\
&= \gamma^k g_n^T S^k r_0 - (-\lambda)^k,
\end{aligned}$$

from which $g_n^T S^k r_0 = (-\tau)^k$. Multiplying (2.6) from the right by the pseudoinverse $[r_0, Sr_0, \ldots, S^n r_0]^+$ and using the fact that g_n lies in the range of $[r_0, Sr_0, \ldots, S^n r_0]$ proves (2.2). Then (2.3) follows in an obvious way. To show (2.4), we denote the N by $n+1$ matrix on the left-hand side and the vector on the right-hand side of (2.6) by R and t, respectively. Then, using (2.5),

$$\begin{aligned}
\|r_n\| \;=\; \|g_n\|^{-1} &\leq \|R\| \, \|t\|^{-1} \\
&\leq \; \|R\|_F \, \|t\|^{-1} \\
&\leq (n+1)^{\frac{1}{2}} \|r_0\| \, \|t\|^{-1},
\end{aligned}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. ∎

Writing (2.6) for the maximal $n = N - l$ in a transposed form gives the upper triangular system for the nonzero entries of $g_{N-l} = [0, \ldots, 0, \chi_l, \chi_{l+1}, \ldots, \chi_N]$,

$$\begin{bmatrix} \rho_l & \rho_{l+1} & \cdots & \rho_N \\ & \rho_l & \cdots & \rho_{N-1} \\ & & \ddots & \vdots \\ & & & \rho_l \end{bmatrix} \begin{bmatrix} \chi_l \\ \chi_{l+1} \\ \vdots \\ \chi_N \end{bmatrix} = \begin{bmatrix} 1 \\ -\tau \\ \vdots \\ (-\tau)^{N-l} \end{bmatrix}. \tag{2.7}$$

The identity (2.5) now immediately implies the following.

COROLLARY 2.2. *With the assumptions and notation of Theorem* 2.1*,*

$$r_{N-l} = \|r_{N-l}\|^2 \, g_{N-l} \quad \textit{and} \quad \|r_{N-l}\| = \|g_{N-l}\|^{-1}, \tag{2.8}$$

where the nonzero entries of g_{N-l} are determined from (2.7) *by back substitution.*

Theorem 2.1 and Corollary 2.2 show how the GMRES residuals depend on J (particularly on the ratio of λ and γ) *and* the structure of r_0. The bound (2.4) is interesting for large values of $|\tau|$ only, i.e., for diagonally dominant matrices J. In the following examples we give explicit formulas for the nth GMRES residual and its norm for some specific initial residuals.

Example 2.3. Suppose that $r_0 = e_l$ is the lth standard basis vector. Then for $n = 0, 1, \ldots, N - l$, $[r_0, Sr_0, \ldots, S^n r_0] = [e_l, e_{l+1}, \ldots, e_{l+n}]$. Hence (2.2) yields

$$r_n^T = \|r_n\|^2 \, [0, \ldots, 0, 1, -\tau, \ldots, (-\tau)^n, 0, \ldots, 0] \,,$$

where r_n^T has $l - 1$ leading and $N - n - l$ trailing zeros, respectively. Taking norms on both sides shows that

$$\|r_n\| = \left(\sum_{j=0}^{n} |\tau|^{2j} \right)^{-\frac{1}{2}}, \tag{2.9}$$

i.e., that equality holds in (2.3) with $\sigma_{\min}([r_0, Sr_0, \ldots, S^n r_0]) = 1$. We see that for $r_0 = e_l$, the GMRES residual norms suffer from slow convergence until the very last step whenever $|\tau| \leq 1$. In their unpublished note [2], Eiermann and Ernst give a proof of (2.9) as well as a slightly weaker form of (2.4) based on a formula for the GMRES minimizing polynomial. They also point out that (2.9) is equivalent to the identity

$$\min_{p \in \pi_n} \left\{ \sum_{j=0}^{n} \left| \frac{p^{(j)}(\tau)}{j!} \right|^2 \right\} = \left(\sum_{j=0}^{n} |\tau|^{2j} \right)^{-1},$$

where $p^{(j)}(\tau)$ denotes the jth derivative of the polynomial $p(\tau)$. This can be of interest independent of the GMRES context. ∎

Example 2.4. Consider the particular case $r_0 = e \equiv [1, 1, \ldots, 1]^T$. Then for $n = 1, 2, \ldots, N - 1$,

$$[e, Se, \ldots, S^n e]^+ = \left[e_1, \, -e_1 + e_2, \, \ldots, \, -e_{n-1} + e_n, \, -e_n + \frac{1}{N-n} S^n \, e \right]^T,$$

which can easily be verified using the four Moore–Penrose conditions; see, e.g., [11, p. 102]. The GMRES residuals are therefore given by

$$\frac{r_n^T}{\|r_n\|^2} = [1, -\tau, \ldots, (-\tau)^n]\,[e, Se, \ldots, S^n e]^+$$
$$= \left[1+\tau, -(\tau+\tau^2), \ldots, (-1)^{n-1}(\tau^{n-1}+\tau^n), \frac{(-\tau)^n}{N-n}, \ldots, \frac{(-\tau)^n}{N-n}\right],$$

and hence

$$\|r_n\| \;=\; \left(|1+\tau|^2 \sum_{k=0}^{n-1} |\tau|^{2k} \;+\; \frac{|\tau|^{2n}}{N-n} \right)^{-\frac{1}{2}}.$$

Similarly to the case $r_0 = e_l$, the GMRES residual norms converge for $r_0 = e$ slowly until the very last step whenever $|\tau| \le 1$. Unlike in the case $r_0 = e_l$, for $r_0 = e$ the GMRES convergence depends on the sign of the real part of τ. In particular,

$$\|r_n\| \;=\; \left(\tfrac{N-n}{4n(N-n)+1}\right)^{\frac{1}{2}} \qquad \text{for } \tau = 1,$$

$$\|r_n\| \;=\; (N-n)^{\frac{1}{2}} \qquad \text{for } \tau = -1.$$

Thus the stagnation is more severe when $\tau = -1$ (recall that $\|r_0\| = N^{\frac{1}{2}}$). ∎

These examples demonstrate that if $|\tau| \le 1$, then slow convergence of the GMRES residual norms can typically be expected.

3. Tridiagonal Toeplitz matrices. Given nonzero parameters γ, λ, and μ, consider an N by N tridiagonal Toeplitz matrix T,

$$T \;=\; \gamma S + \lambda I + \mu S^T \;\equiv\; \gamma\,(S + \tau I + \zeta S^T), \qquad \tau \;\equiv\; \frac{\lambda}{\gamma}, \quad \zeta \;\equiv\; \frac{\mu}{\gamma}. \tag{3.1}$$

Adding a nonzero superdiagonal μS^T to J in (2.1) causes the resulting matrix T to have N distinct eigenvalues,

$$\sigma_k \;=\; \lambda + \mu\zeta^{-\frac{1}{2}}\,\omega_k\,, \qquad \omega_k \;\equiv\; 2\cos\frac{k\pi}{N+1}\,, \qquad k = 1, \ldots, N, \tag{3.2}$$

with the corresponding normalized eigenvectors given by

$$y_k \;=\; \nu_k\,[\Delta\,u_k]\,, \qquad k = 1, \ldots, N, \tag{3.3}$$

where

$$u_k = \left(\frac{2}{N+1}\right)^{\frac{1}{2}} \left[\sin\frac{k\pi}{N+1}, \ldots, \sin\frac{Nk\pi}{N+1}\right]^T,$$
$$\Delta = \operatorname{diag}\left(\zeta^{-\frac{1}{2}}, \zeta^{-1}, \ldots, \zeta^{-\frac{N}{2}}\right),$$
$$\nu_k = \left(\frac{2}{N+1}\sum_{j=1}^{N} \zeta^{-j}\sin^2\frac{jk\pi}{N+1}\right)^{-\frac{1}{2}};$$

see, e.g., [10, pp. 113–115]. Please note that the matrix $U = [u_1, \ldots, u_N]$ represents the real orthonormal and symmetric eigenvector matrix of any N by N *symmetric* (possibly complex) tridiagonal Toeplitz matrix. The eigenvector matrix $Y = [y_1, \ldots, y_N]$

of T is, apart from the normalization, obtained from U by scaling the rows by the powers of $\zeta^{-\frac{1}{2}}$. Hence the condition number of Y equals $\max(|\zeta|^{\frac{1-N}{2}}, |\zeta|^{\frac{N-1}{2}})$.

When $|\gamma| \approx |\mu|$, meaning $|\zeta| \approx 1$, then Y is well-conditioned and one may base the GMRES convergence analysis on the eigenvalues of T *and* the components of r_0 in the direction of the individual eigenvectors of T.

This paper is motivated by the application of GMRES to convection-diffusion problems with dominating convection [7]. Then the interesting case is characterized by $|\gamma| \approx |\lambda| \gg |\mu|$, meaning $|\tau| \approx 1$ and $|\zeta| \ll 1$. The principal question is, To what extent does the behavior of the GMRES residual for T and a given r_0 resemble the behavior of the GMRES residual for the corresponding J and the same r_0? We focus on this question but we also present some general statements valid for arbitrary nonzero values of γ, λ, and μ.

We would like to stress the following subtle point: When $|\zeta|$ is small, the matrix T can be viewed as a small perturbation of the matrix J. It is therefore tempting to conclude that for each given r_0 the Krylov subspaces generated by T and J are in some sense close to each other. This would imply that generally the GMRES residual norms for J and r_0 are close to the GMRES residual norms for T and r_0. However, it is well known that a small perturbation of a general matrix does not ensure a small change of the Krylov subspace, not even when the matrix is symmetric positive definite. (An instructive example is given below.) It is the structure of J and T that makes such arguments applicable and our analysis possible.

3.1. Explicit mapping. The standard approach to GMRES convergence analysis is based on the eigendecomposition $T = YDY^{-1}$, $D = \mathrm{diag}\,(\sigma_1, \ldots, \sigma_N)$, giving

$$\|r_n\| = \|Yp_n(D)Y^{-1}r_0\| = \min_{p\in\pi_n} \|Yp(D)Y^{-1}r_0\| \tag{3.4}$$

$$\leq \|Y\|\,\|Y^{-1}\|\,\|r_0\| \min_{p\in\pi_n} \max_k |p(\sigma_k)|; \tag{3.5}$$

see [3, Theorem 5.4] and [9, Proposition 4]. The resulting worst-case bound (3.5) frequently is the basis for discussions of GMRES convergence. However, it does not take into account the fact that for some initial residuals GMRES may behave very differently than for others. In practical problems we work with some particular initial residuals and we are rarely interested in the worst-case behavior. Moreover, when the eigenvector matrix Y is ill-conditioned, then some components of the vector $Y^{-1}r_0$ can be very large, potentially much larger than $\|r_0\|$. On the other hand, the norm of the linear combination $Y\,[p_n(D)Y^{-1}r_0]$ in (3.4) is bounded from above by $\|r_0\|$. This linear combination therefore can contain a significant cancellation, which is not reflected in the minimization problem (3.5). Hence the principal weakness of (3.5) in case of ill-conditioned eigenvectors is not the potentially large multiplicative factor $\|Y\|\,\|Y^{-1}\|$, in our case equal to $\max(|\zeta|^{\frac{1-N}{2}}, |\zeta|^{\frac{N-1}{2}})$. The principal weakness is rather the minimization problem itself. In general, any description of GMRES convergence using the possibly large coordinates $Y^{-1}r_0$ of r_0 in the eigenvector basis, and the mapping from $Y^{-1}r_0$ to the nth GMRES residual r_n, should be applied with proper care for the cancellation that might occur in the presence of close eigenvectors. For more discussion on this topic, see [7] and [12]. In the following we will show the difference when the mapping from $Y^{-1}r_0$ to r_n is replaced by the mapping from r_0 to r_n.

Let us examine the identity

$$r_n = p_n(T)\,r_0 = \Delta\,U p_n(D) U \Delta^{-1} r_0\,. \tag{3.6}$$

We interpret $p_n(T)$ as the mapping from r_0 to r_n, and we denote, for simplicity, $p_n(T) = C_n$. The entries $c_n^{(jk)}$ of C_n, $j,k = 1,2,\ldots,N$, are given by

$$c_n^{(jk)} \;=\; e_j^T C_n\, e_k \;=\; e_j^T \Delta\, U p_n(D) U \Delta^{-1} e_k \;=\; \zeta^{\frac{k-j}{2}} u_j^T p_n(D)\, u_k\,. \tag{3.7}$$

The jth entry of r_n can be expressed as

$$e_j^T r_n \;=\; e_j^T C_n r_0 \;=\; \sum_{k=1}^{N} c_n^{(jk)} \rho_k. \tag{3.8}$$

Note that since T is tridiagonal, the matrices T^n and thus the matrices C_n, for $n = 0,1,\ldots,N-1$, in general have exactly n nonzero subdiagonals and n nonzero superdiagonals. In particular, $c_n^{(jk)} = 0$ for $|j-k| > n$.

THEOREM 3.1. *For each n until GMRES terminates the mapping C_n from r_0 to r_n represents a banded matrix with $2n+1$ nonzero diagonals. We denote the column vectors formed by the entries of each diagonal (ordered from the most outer subdiagonal to the most outer superdiagonal) by*

$$c_n^{(-n)}, c_n^{(-n+1)}, \ldots, c_n^{(0)}, \ldots, c_n^{(n-1)}, c_n^{(n)}.$$

Then the subdiagonals and superdiagonals are related by

$$c_n^{(d)} = \zeta^d\, c_n^{(-d)}\,, \tag{3.9}$$

and the nth GMRES residual can therefore be written in the form

$$r_n \;=\; C_n r_0 \;=\; \sum_{d=0}^{n} [S^d r_0] \odot \begin{bmatrix} 0_d \\ c_n^{(-d)} \end{bmatrix} \;+\; \zeta \sum_{d=1}^{n} \zeta^{d-1} [(S^T)^d r_0] \odot \begin{bmatrix} c_n^{(-d)} \\ 0_d \end{bmatrix}, \tag{3.10}$$

where $a \odot b$ denotes the element-by-element multiple (Hadamard product) of the vectors a and b, and 0_d denotes the zero vector of length d.

Proof. For a given n, and d fixed between 1 and n, the vector $c_n^{(-d)}$ representing the dth subdiagonal consists of the entries $c_n^{(j,j-d)}$, $j = d+1,\ldots,N$. The manipulations

$$\begin{aligned} c_n^{(j,j-d)} &= \zeta^{-\frac{d}{2}}\, u_j^T p_n(D)\, u_{j-d} \\ &= \zeta^{-d}\, (\, \zeta^{\frac{d}{2}}\, u_{j-d}^T\, p_n(D)\, u_j\,) \\ &= \zeta^{-d} c_n^{(j-d,j)} \end{aligned}$$

finish the proof of (3.9). Relation (3.10) is an obvious consequence of (3.9). ∎

When $|\zeta| \ll 1$, the strictly upper triangular part of the mapping C_n is much less significant than its lower triangular part (including the main diagonal). The significance of the superdiagonals is exponentially decreasing with the distance from the main diagonal. Since the proof of Theorem 3.1 does not use that p_n is the GMRES polynomial, the statement can be reformulated for any matrix polynomial $p(T)$, where T is a tridiagonal Toeplitz matrix.

Using (3.8),

$$\|r_n\|^2 \;=\; \sum_{j=1}^{N} |e_j^T r_n|^2 \;=\; \sum_{j=1}^{N} \left| \sum_{k=1}^{N} c_n^{(jk)} \rho_k \right|^2 . \tag{3.11}$$

Since $C_0 = I$, this formula for $n = 0$ reduces to

$$\|r_0\|^2 \;=\; \sum_{j=1}^{N} \left| \sum_{k=1}^{N} c_0^{(jk)} \rho_k \right|^2 \;=\; \sum_{k=1}^{N} |\rho_k|^2 \,. \tag{3.12}$$

A comparison of (3.11) and (3.12) shows that the decrease of the GMRES residual norms is controlled by the behavior of the individual entries $c_n^{(jk)}$ defined in (3.7). Moreover,

$$\|r_n\| \;\le\; \|r_0\| \, \|C_n\| \;\le\; \|r_0\| \, \|C_n\|_F \,. \tag{3.13}$$

These bounds are different from the usual worst-case convergence bounds in that C_n is determined by p_n, which depends on the particular r_0.

The individual entries of the matrices C_n do not decrease monotonically, but their behavior is typically very different from the behavior of the entries of the mapping $Y p_n(D)$ from $Y^{-1} r_0$ to r_n in (3.4). We do not quantify this in a statement but instead present a qualitative argument and experiments. The only term that can seemingly make $c_n^{(jk)}$ large is $\zeta^{\frac{k-j}{2}}$. When, e.g., $|\zeta| \ll 1$, then for $j > k$ this factor becomes large. However, $c_n^{(jk)}$ are the entries of the matrix $C_n = p_n(T)$. Therefore we may expect that the individual nonzero $c_n^{(jk)}$ are of moderate size, and mostly decreasing (although possibly very slowly) with n, which makes the inequalities (3.13) reasonable. The fact that each iteration step n introduces a new nonzero subdiagonal in the mapping from r_0 to r_n hints that when $|\gamma| \approx |\lambda| \gg |\mu|$, i.e., $|\tau| \approx 1 \gg |\zeta|$, the GMRES convergence may be slow.

We emphasize that these considerations about C_n and convergence of GMRES are based on the particular tridiagonal Toeplitz structure of T. On the other hand, when the components of $Y^{-1} r_0$ are large, any approach based on $Y p_n(D)$ can hardly lead to a well-justified insight, even when the special structure of T is exploited.

In Figure 3.1 we plot the values $\log_{10}(|c_n^{(jk)}|)$, $j, k = 1, \ldots, 15$, for $n = 2$, 6, 10, 14, computed when GMRES is applied to the 15 by 15 matrix $T_1 = S + I + 0.01\, S^T$ and the initial residual $r_0 = e$. Corresponding results for T_1 and $r_0 = \texttt{rand}(15, 1)$ are shown in Figure 3.2 (`rand` is the pseudorandom number generator in MATLAB), and Figure 3.3 shows results for the diagonally dominant matrix $T_2 = S + 2I + 0.01\, S^T$ and $r_0 = e$. In Figure 3.4 we plot the respective GMRES residual norms and in Figure 3.5 the values $\|C_n\|_F$, representing an upper bound on $\|r_n\|/\|r_0\|$; cf. (3.13).

In Figures 3.1 to 3.3 we see a decrease of $|c_n^{(jk)}|$ on the superdiagonals of C_n that is exponential in the distance from the main diagonal. Hence in the individual sums $|\sum_{k=1}^{N} c_n^{(jk)} \rho_k|^2$, $j = 1, \ldots, N$, on the right-hand side of (3.11) only the terms for $j \ge k$ play a significant role.

For T_1 and $r_0 = e$ as well as $r_0 = \texttt{rand}(15, 1)$, the significant entries $c_n^{(jk)}$ maintain approximately the same orders of magnitude throughout the GMRES iteration. Correspondingly, the residual norms (solid and dash-dot curves in Figure 3.4) decrease very slowly until the very last step. The initial residual $r_0 = e$ presents a case that yields almost a perfect plateau of significant $|c_n^{(jk)}|$ in every step. The variation of the entries in $r_0 = \texttt{rand}(15, 1)$ causes a larger variation among the absolute values of the significant entries of C_n. For T_2 and $r_0 = e$, GMRES converges faster (cf. the dashed curve in Figure 3.4) since all significant entries of C_n decrease noticeably in magnitude in every step. A comparison of Figure 3.4 and Figure 3.5 illustrates that

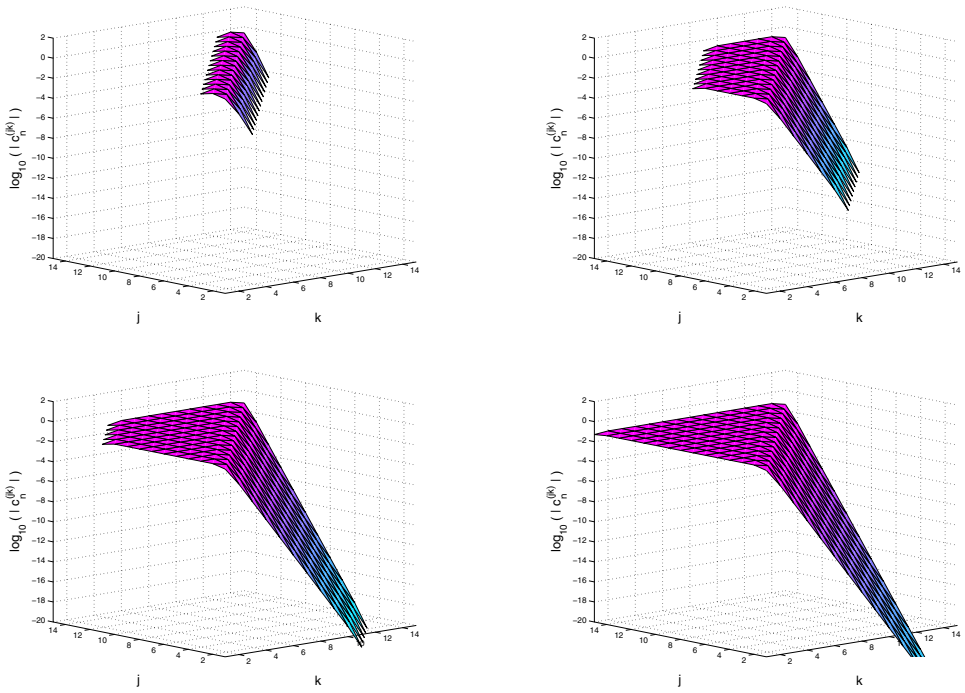


FIG. 3.1. *The values* $\log_{10}(|c_n^{(jk)}|)$ *for* $j,k = 1,\ldots,N$ *and* $n = 2$ *(top left),* 6 *(top right),* 10 *(bottom left),* 14 *(bottom right), computed when GMRES is applied to the* 15 *by* 15 *matrix* $T_1 = S + I + 0.01\,S^T$ *and* $r_0 = e$.

the inequalities (3.13) are for our data quite sharp, and, consequently, that there is no significant cancellation among the individual terms in (3.11).

In the following subsection we develop an analogue of Theorem 2.1 for tridiagonal Toeplitz matrices.

3.2. Extension of the bidiagonal analysis. For each scaled (lower bidiagonal) Jordan block J and each r_0 it is easy to see when GMRES terminates: if ρ_l is the first nonzero entry of r_0, then GMRES applied to J and r_0 terminates in exactly $N - l + 1$ steps, giving $r_{N-l} \neq 0$ and $r_{N-l+1} = 0$. For a tridiagonal Toeplitz matrix T with nonzero sub- and superdiagonal, the situation is more complicated. Here the total number of GMRES steps for a given nonzero pattern of r_0 can depend on the actual numerical values of its nonzero entries. However, since we are not interested in conditions for termination of GMRES in a given number of steps, we will not specify this number and merely assume that it is greater than $N - l$.

THEOREM 3.2. *Suppose that GMRES is applied to a system with the matrix* $T = \gamma(S + \tau I + \zeta S^T)$ *and the initial residual* $r_0 = [\rho_1,\ldots,\rho_N]^T$. *Let* ρ_l *be the first nonzero entry of* r_0. *Moreover, suppose that* r_0 *has at least* $N - l$ *nonzero components in the directions of the individual eigenvectors of the matrix* T *(GMRES does not terminate in the first* $N - l$ *steps). Then for* $n = 0, 1, \ldots, N - l$ *the GMRES residuals satisfy*

$$r_n^T = \|r_n\|^2\,[1, -\tau, \ldots, (-\tau)^n]\,[r_0, (S + \zeta S^T)r_0, \ldots, (S + \zeta S^T)^n r_0]^+\,, \tag{3.14}$$

$$\|r_n\| \geq \left(\sum_{j=0}^{n} |\tau|^{2j}\right)^{-\frac{1}{2}} \sigma_{\min}\left([r_0, (S + \zeta S^T)r_0, \ldots, (S + \zeta S^T)^n r_0]\right). \tag{3.15}$$

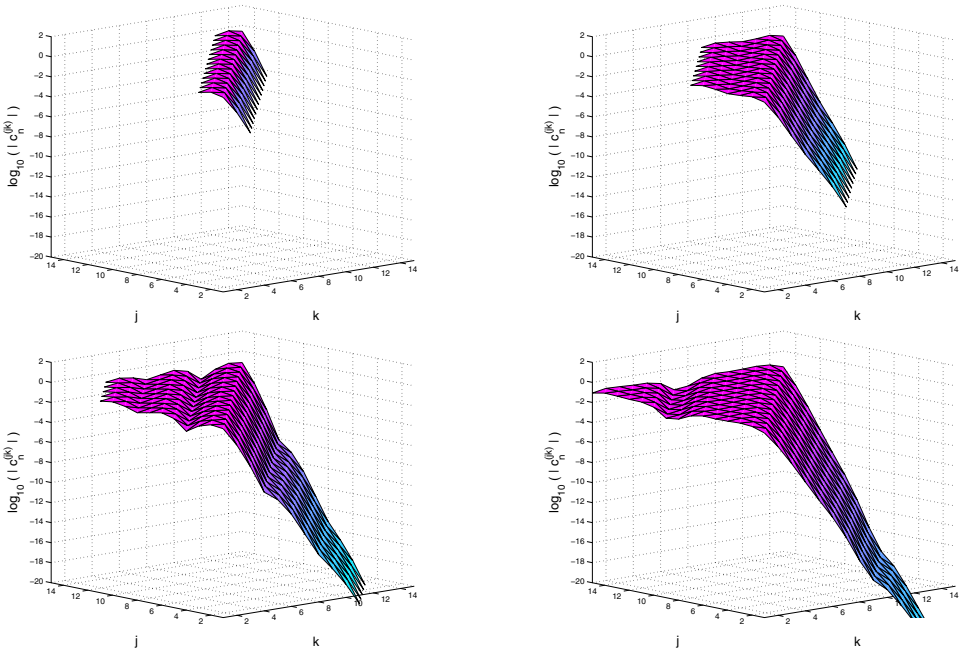


FIG. 3.2. *The values* $\log_{10}(|c_n^{(jk)}|)$ *for* $j,k = 1,\ldots,N$ *and* $n = 2$ *(top left),* 6 *(top right),* 10 *(bottom left),* 14 *(bottom right), computed when GMRES is applied to the* 15 *by* 15 *matrix* $T_1 = S + I + 0.01\,S^T$ *and* $r_0 = \texttt{rand}(15,1)$.

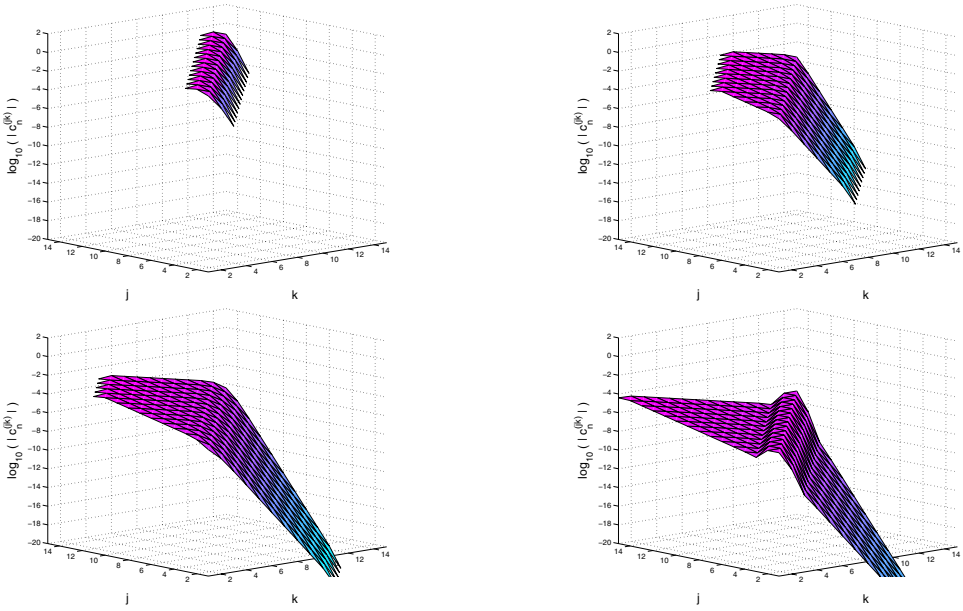


FIG. 3.3. *The values* $\log_{10}(|c_n^{(jk)}|)$ *for* $j,k = 1,\ldots,N$ *and* $n = 2$ *(top left),* 6 *(top right),* 10 *(bottom left),* 14 *(bottom right), computed when GMRES is applied to the* 15 *by* 15 *matrix* $T_2 = S + 2I + 0.01\,S^T$ *and* $r_0 = e$.

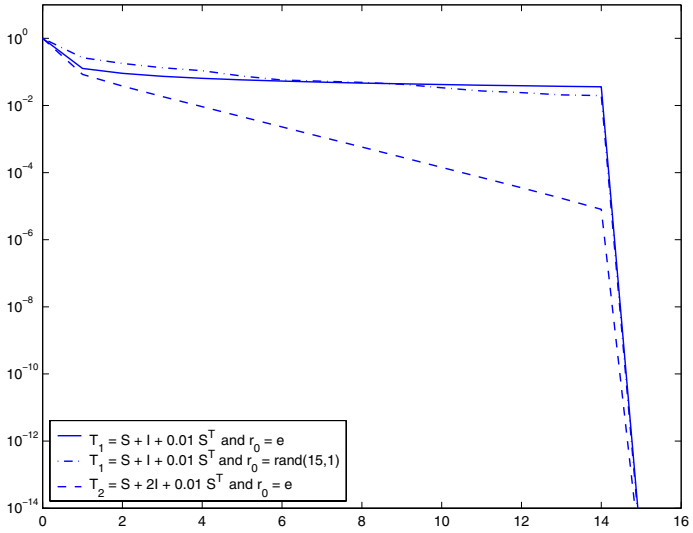


FIG. 3.4. *Residual norms* $\|r_n\|/\|r_0\|$ *of GMRES applied to* $T_1 = S + I + 0.01\,S^T$ *and* $r_0 = e$ *(solid),* T_1 *and* $r_0 = \mathtt{rand}(15,1)$ *(dash-dot),* $T_2 = S + 2I + 0.01\,S^T$ *and* $r_0 = e$ *(dashed).*

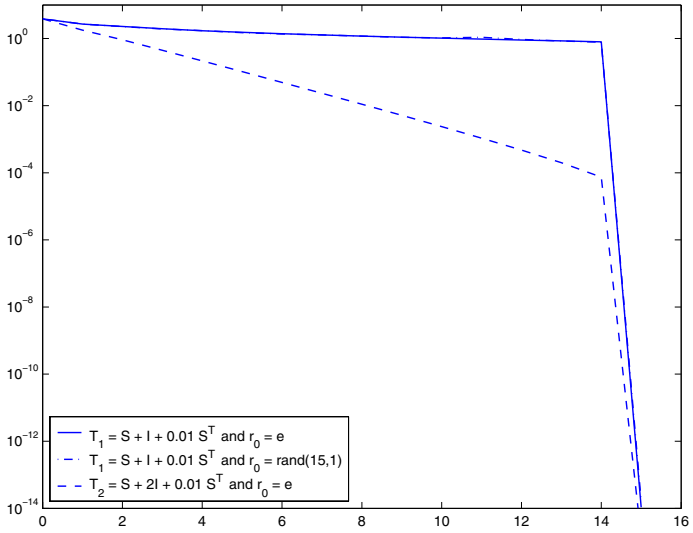


FIG. 3.5. *The values* $\|C_n\|_F$ *for* $T_1 = S+I+0.01\,S^T$ *and* $r_0 = e$ *(solid),* T_1 *and* $r_0 = \mathtt{rand}(15,1)$ *(dash-dot),* $T_2 = S + 2I + 0.01\,S^T$ *and* $r_0 = e$ *(dashed).*

Proof. For $n = 0, 1, \ldots, N-l$, the matrix $[r_0, Tr_0, \ldots, T^n r_0]$ has full column rank. The rest is similar to the proof of Theorem 2.1, with T and $S + \zeta S^T$ taking over the roles of J and S, respectively. Indeed,

$$r_n^T \;=\; \|r_n\|^2\, e_1^T\, [r_0, Tr_0, \ldots, T^n r_0]^+ \;\equiv\; \|r_n\|^2\, g_n^T\,,$$

from which we receive $g_n^T\,[r_0, Tr_0, \ldots, T^n r_0] = e_1^T$. Then

$$0 \;=\; g_n^T T r_0 \;=\; \gamma g_n^T\,(S + \zeta S^T)\, r_0 \;+\; \lambda g_n^T r_0, \qquad \text{i.e., } g_n^T\,(S + \zeta S^T)\, r_0 \;=\; -\tau,$$

and an induction shows that in fact $g_n^T (S+\zeta S^T)^k r_0 = (-\tau)^k$ for $k = 1, 2, \ldots$. Hence

$$g_n^T \left[r_0, (S+\zeta S^T) r_0, \ldots, (S+\zeta S^T)^n r_0 \right] \;=\; \left[1, -\tau, \ldots, (-\tau)^n \right]. \tag{3.16}$$

Now note that

$$g_n \in \mathrm{span}\{r_0, Tr_0, \ldots, T^n r_0\} = \mathrm{span}\{r_0, (S+\zeta S^T) r_0, \ldots, (S+\zeta S^T)^n r_0\}.$$

A multiplication of (3.16) from the right with

$$[r_0, (S+\zeta S^T) r_0, \ldots, (S+\zeta S^T)^n r_0]^+$$

yields (3.14). The lower bound (3.15) is a direct consequence. □

Consider, for simplicity, the iteration step $n = N - l$. The principal difference between the cases with J and T is in the form of (2.7) and (3.16). The system of equations (3.16) is for $l \neq 1$ underdetermined, and its system matrix is constructed from r_0 in a much more complicated way than in (2.7). However, the system matrix in (3.16) can be written in the form

$$\begin{aligned} &[r_0, S r_0, \ldots, S^{N-l} r_0]^T + \zeta \left[0, S^T r_0, \ldots, \zeta^{-1} \left\{ (S+\zeta S^T)^{N-l} - S^{N-l} \right\} r_0 \right]^T \\ &\qquad\qquad \equiv [O, R] + \zeta P, \end{aligned} \tag{3.17}$$

where O denotes the $N-l+1$ by $l-1$ zero matrix, and R denotes the upper triangular matrix described in (2.7). The columns of P^T are given by

$$p_j \;=\; \zeta^{-1} \left\{ (S+\zeta S^T)^j - S^j \right\} r_0 \qquad \text{for } j = 0, 1, \ldots, N-l. \tag{3.18}$$

Since S and S^T do not commute, $(S+\zeta S^T)^j$ cannot be evaluated by the binomial theorem. However, for $j = 1, \ldots, N-l$, this expression can be formally written as

$$(S+\zeta S^T)^j \;=\; \Sigma_{j,0} + \zeta \Sigma_{j,1} + \cdots + \zeta^{j-1} \Sigma_{j,j-1} + \zeta^j \Sigma_{j,j}.$$

Here $\Sigma_{j,k}$ denotes the sum of all possible matrix products involving $j-k$ times the matrix S and k times the matrix S^T. In particular, $\Sigma_{j,0} = S^j$ and $\Sigma_{j,j} = (S^T)^j$. Consequently, for $j = 1, \ldots, N-l$,

$$p_j \;=\; \left(\Sigma_{j,1} + \zeta \Sigma_{j,2} + \cdots + \zeta^{j-2} \Sigma_{j,j-1} + \zeta^{j-1} \Sigma_{j,j} \right) r_0 \,. \tag{3.19}$$

Note that the matrix $\Sigma_{j,k}$ is, for $1 \le j \le N-l$ and $1 \le k \le j$, the sum of $\binom{j}{k}$ products of shift matrices and that $\|\Sigma_{j,k}\| \;\le \binom{j}{k}$. Therefore, assuming $|\zeta| \ll (j-1)^{-1}$,

$$\|p_j\| \;\le\; \|r_0\| \sum_{k=1}^{j} |\zeta|^{k-1} \begin{pmatrix} j \\ k \end{pmatrix} \;=\; j \, \|r_0\| \, (1 + \mathcal{O}(|\zeta| \, j)),$$

where $\mathcal{O}(z)$ is bounded from above by z multiplied by a constant (here close to one). When $|\zeta| \;\ll\; (N-l)^{-\frac{3}{2}}$,

$$\|P\| \;\le\; (N-l)^{\frac{1}{2}} \max_j \|p_j\| \;\le\; (N-l)^{\frac{3}{2}} \, \|r_0\| \left(1 + \mathcal{O}\left((N-l)^{-\frac{1}{2}} \right) \right).$$

The matrix (3.17) can then be considered a small perturbation of the upper triangular system matrix in (2.7), extended by a zero block.

We will now use this perturbation idea for analyzing when GMRES applied to T and r_0 behaves similarly to GMRES applied to J and r_0. As mentioned above, this phenomenon depends in a complicated way on the initial residual r_0; cf. (3.16) and (3.17). Any general result with a nontrivial quantitative meaning can therefore be expected to reflect this complicated nature. In the following we have chosen to preserve a quantitative character of the bounds at the price of an assumption on $R^{-1}P$.

We will use the following notation. The residual for GMRES applied to J with r_0 and the auxiliary vector obtained as a solution of (2.7) will be denoted by $r_n^{(J)}$ and $g_n^{(J)}$, respectively. Analogously, $r_n^{(T)}$, respectively, $g_n^{(T)}$, will denote the residual for GMRES applied to T with r_0, respectively, the minimum norm solution of (3.16). As above, let $r_0 = [\rho_1, \ldots, \rho_N]^T$ with ρ_l being its first nonzero entry. As in Theorem 3.2 we will assume that GMRES applied to T with r_0 does not terminate in the first $N-l$ steps. Then from (3.16),

$$\begin{aligned} g_{N-l}^{(T)} &= ([O,R] + \zeta P)^+ \, [1, -\tau, \ldots, (-\tau)^{N-l}]^T \\ &= \left([O,I] + \zeta R^{-1}P\right)^+ R^{-1}[1, -\tau, \ldots, (-\tau)^{N-l}]^T \\ &= \left([O,I] + \zeta R^{-1}P\right)^+ g_{N-l}^{(J)} \,. \end{aligned}$$

Taking norms,

$$\|[O,I] + \zeta R^{-1}P\|^{-1} \, \|g_{N-l}^{(J)}\| \;\le\; \|g_{N-l}^{(T)}\| \;\le\; \|([O,I] + \zeta R^{-1}P)^+\| \, \|g_{N-l}^{(J)}\| \,. \tag{3.20}$$

Assuming that $|\zeta| \, \|R^{-1}P\| < 1$,

$$\|([O,I] + \zeta R^{-1}P)^+\| \;\le\; (1 - |\zeta| \, \|R^{-1}P\|)^{-1} \,.$$

Considering that $\|r_{N-l}^{(T)}\| = 1/\|g_{N-l}^{(T)}\|$ and $\|r_{N-l}^{(J)}\| = 1/\|g_{N-l}^{(J)}\|$, we proved the following theorem.

THEOREM 3.3. *Using the previous notation and the assumptions of Theorem* 3.2, *let* $|\zeta| \, \|R^{-1}P\| < 1$. *Then the GMRES residuals* $r_{N-l}^{(T)}$ *and* $r_{N-l}^{(J)}$ *satisfy the inequalities*

$$\left(1 + |\zeta| \, \|R^{-1}P\|\right) \|r_{N-l}^{(J)}\| \;\ge\; \|r_{N-l}^{(T)}\| \;\ge\; \left(1 - |\zeta| \, \|R^{-1}P\|\right) \|r_{N-l}^{(J)}\| \,, \tag{3.21}$$

where R *represents the matrix formed by the last* $N - l + 1$ *columns of the matrix* $[r_0, Sr_0, \ldots, S^{N-l}r_0]^T$ *and* $P = [0, S^T r_0, \ldots, \zeta^{-1}\{(S + \zeta S^T)^{N-l} - S^{N-l}\} \, r_0]^T$.

The main point can be summarized in the following way. Suppose that a scaled Jordan block J is extended to a tridiagonal Toeplitz matrix T by a superdiagonal of sufficiently small modulus (compared to the modulus of the subdiagonal). Assume that GMRES for T and r_0 terminates no earlier than GMRES for J and r_0. Then the convergence of GMRES for T and r_0 will be comparable to the convergence of GMRES for J and r_0. We next consider two examples illustrating our results.

Example 3.4. Suppose that $r_0 = e_1$. Then for J as well as for T the GMRES algorithm terminates in step N. Thus, whenever $|\zeta| \, \|R^{-1}P\| < 1$, the inequalities (3.21) hold with $l = 1$. Note that for $r_0 = e_1$ we have $R = I$ and $\|r_0\| = 1$, so that

$$\begin{aligned} (1 + |\zeta| \, \|P\|) \, \|r_{N-1}^{(J)}\| \;\ge\; \|r_{N-1}^{(T)}\| &\ge (1 - |\zeta| \, \|P\|) \, \|r_{N-1}^{(J)}\| \\ &\ge (1 - |\zeta|(N-1)^{\frac{3}{2}}(1 + \mathcal{O}((N-1)^{-\frac{1}{2}}))) \, \|r_{N-1}^{(J)}\| \,, \end{aligned}$$

when $|\zeta| \;\ll\; (N-1)^{-\frac{3}{2}}$. ∎

Example 3.5. For $r_0 = e \equiv [1, 1, \ldots, 1]^T$ we can see one of the main differences between the application of GMRES to linear systems with J and with a general extension of J to the tridiagonal Toeplitz matrix T: for any nonzero γ and λ, $\dim \mathcal{K}_N(J, e) = N$, and hence GMRES with J and $r_0 = e$ terminates in step N. For certain nonzero values of λ, γ, and μ, however, $\dim \mathcal{K}_N(T, e) < N$, and hence for certain matrices T and $r_0 = e$ the GMRES algorithm terminates earlier than in step N.

The prime example for the latter case is given by a symmetric T, i.e., $\gamma = \mu$. The normalized eigenvectors of each such matrix are given in (3.3) with $\Delta = I$. These vectors represent discrete sine functions and thus they satisfy certain symmetries. In particular, simple technical manipulations show that

$$\begin{aligned} u_k^T\, e &= \left(\frac{2}{N+1}\right)^{\frac{1}{2}} \sum_{j=1}^{N} \sin\left(\frac{jk\pi}{N+1}\right) \\ &= \left(\frac{2}{N+1}\right)^{\frac{1}{2}} \frac{\cos\left(\frac{k\pi}{2(N+1)}\right) - \cos\left(\frac{(2N+1)k\pi}{2(N+1)}\right)}{2\sin\left(\frac{k\pi}{2(N+1)}\right)} \\ &= \left(\frac{2}{N+1}\right)^{\frac{1}{2}} \frac{\cos\left(\frac{k\pi}{2(N+1)}\right)}{2\sin\left(\frac{k\pi}{2(N+1)}\right)} \left(1 - (-1)^k\right) \\ &= 0 \qquad \text{if } k \text{ is even.} \end{aligned}$$

When $u_k^T r_0 = 0$, the initial residual r_0 has no component in the direction of the eigenvector u_k of T. For a symmetric T and $r_0 = e$, GMRES will therefore terminate in step $N/2$ or $(N+1)/2$ when N is even or odd, respectively. A similar result holds for $\gamma = -\mu$.

In general, however, the normalized eigenvectors of a tridiagonal Toeplitz matrix T are given by $\nu_k[\Delta u_k]$. The components of $r_0 = e$ in the direction of the individual eigenvectors of T are generally given by

$$\nu_k^{-1}\,(u_k^T\,\Delta^{-1}\,e) \;=\; \nu_k^{-1}\,\left(\frac{2}{N+1}\right)^{\frac{1}{2}} \sum_{j=1}^{N} \zeta^{\frac{j}{2}}\,\sin\left(\frac{jk\pi}{N+1}\right)$$

for $k = 1, \ldots, N$. If $|\zeta| \neq 1$, then the initial residual $r_0 = e$ usually has a nonzero component in the direction of *each* of the individual eigenvectors of T. This implies that a very small additive perturbation of a symmetric, even positive definite, tridiagonal Toeplitz matrix by ϵS (or by ϵS^T) may cause GMRES (with $r_0 = e$) to iterate twice as long until it terminates.

Here we are mainly interested in the case $|\zeta| \ll 1$. Then GMRES for J and $r_0 = e$, and usually also for T and $r_0 = e$, terminates in step N. If $|\zeta|\,\|R^{-1}P\| < 1$, then (3.21) holds with $l = 1$. Since $R^{-1} = I - S^T$, we get $\|R^{-1}\| \leq 2$, and since $\|r_0\| = N^{\frac{1}{2}}$, the lower bound in (3.21) yields

$$\begin{aligned} \|r_{N-1}^{(T)}\| &\geq (1 - |\zeta|\,\|(I - S^T)P\|)\,\|r_{N-1}^{(J)}\| \\ &\geq (1 - 2|\zeta|N^2(1 + \mathcal{O}((N-1)^{-\frac{1}{2}})))\,\|r_{N-1}^{(J)}\|\,, \end{aligned}$$

when $|\zeta| \ll (N-1)^{-\frac{3}{2}}$. Numerical examples for this bound are given in section 4. ∎

4. Numerical experiments. The numerical experiments in this section illustrate main points presented and discussed above.

Experiment 4.1. We use the 15 by 15 matrices

$$\begin{aligned} J &= S + I, \\ T_1 &= S + I + 0.01\, S^T, \\ T_2 &= S + I + 0.03\, S^T, \\ T_3 &= S + I + 0.05\, S^T, \\ T_4 &= S + I + 0.999\, S^T, \end{aligned} \tag{4.1}$$

and $r_0 = e$. Since $\dim \mathcal{K}_N(J, e) = N$, and $\dim \mathcal{K}_N(T_j, e) = N$ for all j, GMRES with each of the five matrices and $r_0 = e$ terminates in step N. The relevant values for the application of the bound (3.21) are given in the following table:

j	ζ_j	$\|R_j^{-1}P_j\|$	$\zeta_j \|R_j^{-1}P_j\|$	$1-\zeta_j \|R_j^{-1}P_j\|$
1	0.01	25.58	0.256	0.744
2	0.03	29.50	0.885	0.115
3	0.05	34.33	1.716	$\star$
4	0.999	5.1e+04	5.1e+04	$\star$

For $j = 1, 2$, we have $\zeta_j \|R_j^{-1}P_j\| < 1$, so that the bounds (3.21) are applicable with $l = 1$. The $\star$ for $j = 3, 4$ indicates that since $\zeta_j \|R_j^{-1}P_j\| > 1$, the lower bound in (3.21) is not applicable.

Figures 4.1 and 4.2 show the GMRES residual norms. Since $\tau = 1$, GMRES converges slowly when applied to J (solid). For T_1 (dash-dot) and T_2 (dotted), the GMRES residual norms are very close to the ones for J. The correspondence between $\|r_{14}^{(J)}\|$ and $\|r_{14}^{(T_j)}\|$, $j = 1, 2$, is even closer than predicted by the bounds (3.21). It is also noteworthy that although this bound is not applicable for T_3, the residual norms in this case (dots) are very close to the ones for J as well. The results for T_4 (dashed) show that for a larger perturbation (here $\zeta_4 = 0.999$) the $(N-1)$st GMRES residual norm for a tridiagonal Toeplitz matrix can differ significantly from the corresponding one for the Jordan block.

Experiment 4.2. In Figure 4.3 we used the 15 by 15 matrices

$$\begin{aligned} J &= S + I, \\ T_4 &= S + I + 0.999\, S^T \qquad \text{(as in Experiment 4.1)}, \\ T_5 &= S + I + S^T, \end{aligned}$$

and $r_0 = e$. This experiment demonstrates the difference in the GMRES residual norm curves for T_4 (dash-dot) and T_5 (dotted), despite the fact that $T_4 = T_5 - 0.001\, S^T$ is only a small perturbation of the symmetric matrix T_5. It is interesting to observe that until termination of GMRES for T_5 the convergence curves are very close to each other.

Experiment 4.3. Our last experiment comes from the streamline upwind Petrov–Galerkin (SUPG) discretization of a convection-diffusion model problem with dominating convection. This model problem with rectangular domain, regular grid, and a constant grid aligned convection motivated our work, leading to the results presented in this paper. Here we use it for a short illustration.

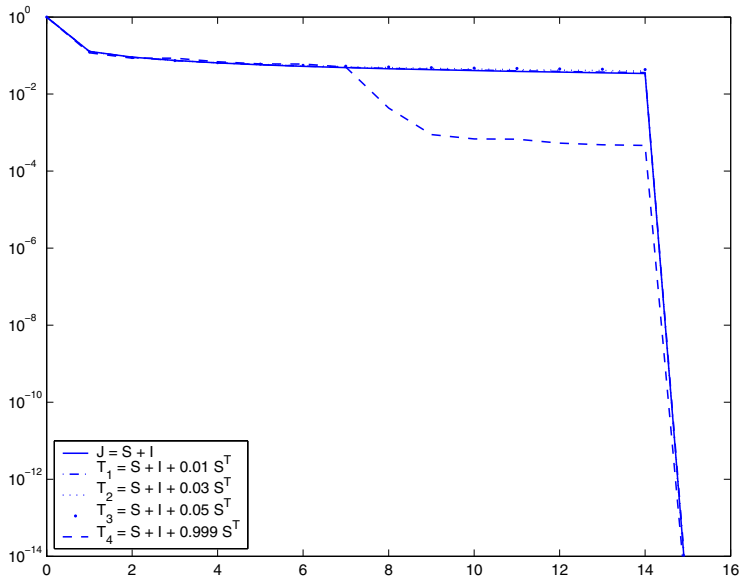


FIG. 4.1. *Residual norms* $\|r_n\|/\|r_0\|$ *of GMRES applied to the five different* 15 *by* 15 *matrices given in* (4.1) *and the initial residual* $r_0 = e$.

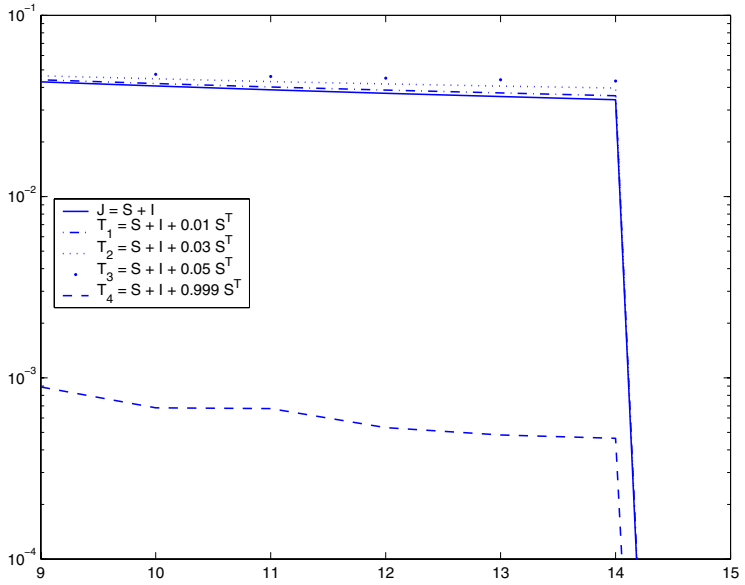


FIG. 4.2. *Close-up of Figure* 4.1.

As explained in [1, 2] and [7], the SUPG discretized model operator can be written as an N^2 by N^2 block-diagonal matrix with N by N nonsymmetric tridiagonal Toeplitz blocks $T_j = \gamma_j(S + \tau_j I + \zeta_j S^T)$, $j = 1, \ldots, N$, on its diagonal. Example values for $|\tau_j|$ and $|\zeta_j|$, as well as the corresponding quantities related to (3.21) with $N = 15$ and $r_0 = e_1$, are given in Table 4.1.

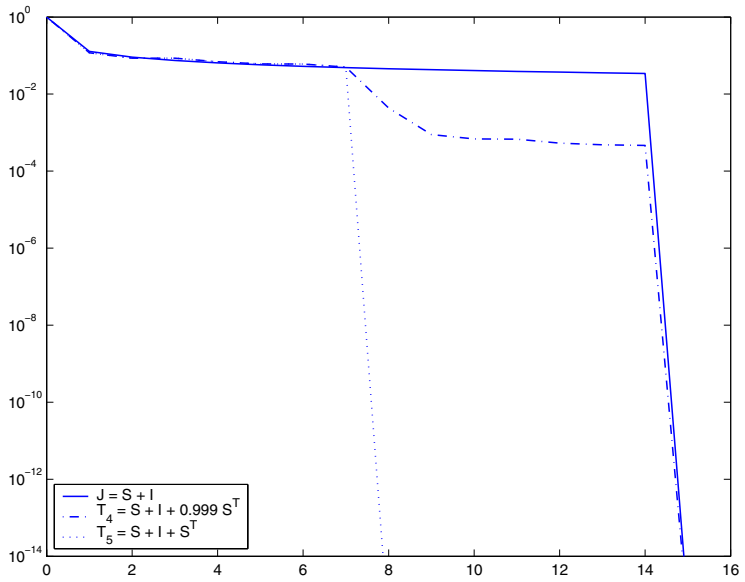


FIG. 4.3. *Residual norms* $\|r_n\|/\|r_0\|$ *of GMRES applied to* 15 *by* 15 *matrices* $J = S + I$ *(solid),* $T_5 = S + I + S^T$ *(dotted),* $T_4 = T_5 - 0.001\, S^T$ *(dash-dot), and the initial residual* $r_0 = e$.

TABLE 4.1
Example values derived from the SUPG discretized convection-diffusion model operator.

j	$\lvert\tau_j\rvert$	$\lvert\zeta_j\rvert$	$\|R_j^{-1}P_j\|$	$\lvert\zeta_j\rvert\,\|R_j^{-1}P_j\|$
1	1.0052	0.0010	13.0002	0.0134
2	1.0209	0.0042	13.0040	0.0544
3	1.0481	0.0096	13.0211	0.1252
4	1.0881	0.0176	13.0708	0.2303
5	1.1431	0.0286	13.1874	0.3774
6	1.2162	0.0432	13.4295	0.5808
7	1.3116	0.0623	13.8989	0.8663
8	1.4348	0.0870	14.7740	1.2847
9	1.5925	0.1185	16.3739	1.9402
10	1.7923	0.1585	19.2798	3.0551
11	2.0409	0.2082	24.5496	5.1108
12	2.3392	0.2678	34.0035	9.1077
13	2.6735	0.3347	50.1498	16.7855
14	3.0033	0.4007	74.1263	29.6989
15	3.2564	0.4513	99.9102	45.0870

Figure 4.4 shows the GMRES residual norm curves for the matrices T_j, $j = 1, \ldots, 15$, and $r_0 = e_1$. For small j we have $|\tau_j| \approx 1$, which leads to very slow convergence of GMRES for the corresponding scaled Jordan blocks and $r_0 = e_1$. Simultaneously there holds $|\zeta_j| \ll 1$, so that the convergence for the respective tridiagonal Toeplitz matrices T_j with the same r_0 is comparably slow. With increasing j, both $|\tau_j|$ and $|\zeta_j|$ increase, and the speed of convergence of GMRES for T_j (as well as for the corresponding Jordan blocks) and $r_0 = e_1$ increases significantly. The slow convergence of GMRES for the matrices T_j with small indices j translates into an initial phase of slow convergence of GMRES for the SUPG discretized model operator. The detailed exposition is beyond the scope of this paper, and we refer an interested reader to [7].

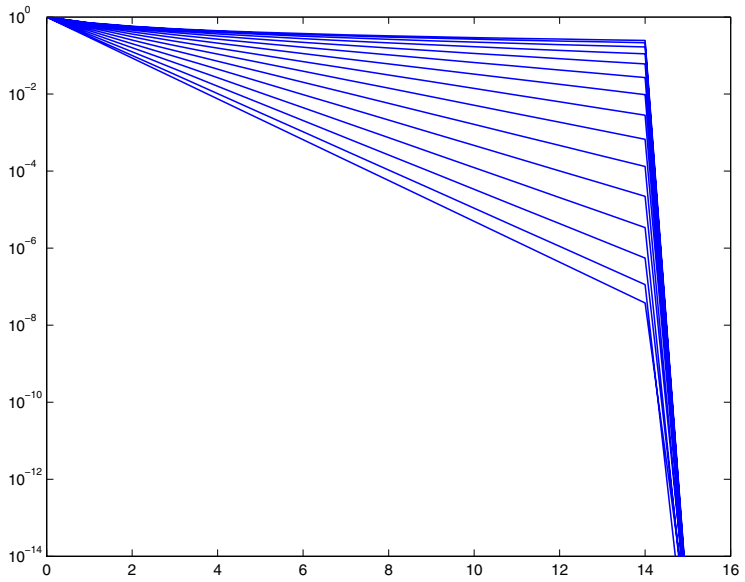


FIG. 4.4. *Residual norms* $\|r_n\|/\|r_0\|$ *of GMRES applied to* 15 *by* 15 *matrices* T_j, $j = 1, \ldots, 15$, *representing the tridiagonal Toeplitz blocks on the diagonal of a SUPG discretized convection-diffusion model operator (see* [7]*) with the initial residuals* $r_0 = e_1$.

5. Conclusions and outlook. Consider GMRES convergence for a matrix A and a given initial residual r_0. Let B be a small perturbation of A. Does the assumption that B is *sufficiently close* to A guarantee that the GMRES residuals for A and r_0 are at every iteration step close to the GMRES residuals for B and r_0? A related question, although in a different context and without the dependence on the initial residual, which we consider vital, was recently also considered by Huhtanen and Nevanlinna [4]. Motivated by applications in convection-diffusion problems [7], our paper studies this question for $A \equiv J = \gamma S + \lambda I$ and $B \equiv T = J + \mu S^T$, and for this particular matrix A and its particular perturbation B it gives an affirmative answer. In general, however, the answer is complicated, which is documented by a nonsymmetric perturbation of a symmetric tridiagonal Toeplitz matrix. To what extent our results can be applied to GMRES convergence analysis of more general problems, e.g., when there exists a well-conditioned transformation of the system matrix into a block diagonal form with tridiagonal blocks, remains the subject of further work.

Acknowledgments. We thank Michael Eiermann and Oliver Ernst for sharing their unpublished notes [2] and for very stimulating discussions and advice about the subject matter of this paper. We also thank the anonymous referee for several suggestions that helped to improve the presentation of the paper. All numerical experiments in this paper were performed using MATLAB [13].

REFERENCES

[1] M. EIERMANN, *Semiiterative Verfahren für nichtsymmetrische lineare Gleichungssysteme*, Habilitationsschrift, Universität Karlsruhe, Karlsruhe, 1989.
[2] M. EIERMANN AND O. ERNST, *GMRES and Jordan blocks,* private communication, 2002.
[3] H. C. ELMAN, *Iterative Methods for Large Sparse Nonsymmetric Systems of Linear Equations*, Ph.D. thesis, Yale University, New Haven, CT, 1982.

[4] M. Huhtanen and O. Nevanlinna, *Minimal decompositions and iterative methods*, Numer. Math., 86 (2000), pp. 257–281.
[5] I. C. F. Ipsen, *Expressions and bounds for the GMRES residual*, BIT, 40 (2000), pp. 524–535.
[6] J. Liesen, M. Rozložník, and Z. Strakoš, *Least squares residuals and minimal residual methods*, SIAM J. Sci. Comput., 23 (2002), pp. 1503–1525.
[7] J. Liesen and Z. Strakoš, *GMRES convergence analysis for a convection-diffusion model problem*, SIAM J. Sci. Comput., submitted.
[8] C. C. Paige and Z. Strakoš, *Residual and backward error bounds in minimum residual Krylov subspace methods*, SIAM J. Sci. Comput., 23 (2002), pp. 1898–1923.
[9] 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.
[10] G. D. Smith, *Numerical solution of partial differential equations*, 2nd ed., Clarendon Press, Oxford, UK, 1978.
[11] G. W. Stewart and J. G. Sun, *Matrix perturbation theory*, Academic Press, Boston, 1990.
[12] L. N. Trefethen, *Pseudospectra of linear operators*, SIAM Rev., 39 (1997), pp. 383–406.
[13] The MathWorks, Inc., *MATLAB* 6.5*, Release* 13, Natick, MA, USA, 2002.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 1, pp. 252–256

NORMWISE SCALING OF SECOND ORDER POLYNOMIAL MATRICES*

HUNG-YUAN FAN†, WEN-WEI LIN†, AND PAUL VAN DOOREN‡

Abstract. We propose a minimax scaling procedure for second order polynomial matrices that aims to minimize the backward errors incurred in solving a particular linearized generalized eigenvalue problem. We give numerical examples to illustrate that it can significantly improve the backward errors of the computed eigenvalue-eigenvector pairs.

Key words. generalized eigenvalues, QZ algorithm, balancing

AMS subject classifications. 15A18, 15A22, 65F15, 65F35

DOI. 10.1137/S0895479803434914

1. Introduction. The quadratic eigenvalue problem (QEP) is the calculation of the roots of the determinant of the polynomial matrix

$$P(\lambda) = \lambda^2 P_2 + \lambda P_1 + P_0, \tag{1.1}$$

where $P_2, P_1, P_0 \in \mathbb{C}^{n\times n}$. A recommended method to solve it, is to reduce it to a generalized eigenvalue problem (GEP), which is the calculation of the roots of the determinant of the following pencil:

$$\lambda B - A = \lambda \begin{bmatrix} I & 0 \\ 0 & P_2 \end{bmatrix} - \begin{bmatrix} 0 & I \\ -P_0 & -P_1 \end{bmatrix}. \tag{1.2}$$

Indeed, one easily verifies that

$$\det(P(\lambda)) \equiv \det(\lambda B - A).$$

But if the matrices P_i, $i = 0, 1, 2$, have norms

$$\gamma_2 := \|P_2\|_2, \quad \gamma_1 := \|P_1\|_2, \quad \gamma_0 := \|P_0\|_2$$

that differ a lot in order of magnitude, then it was shown in [3, Table 5.1] that the QZ algorithm applied to (1.2) may yield very poor backward errors in the coefficients of the polynomial matrix (1.1).

In this note we relate this to the scaling problem of the polynomial matrix (1.1) and we indicate that the computed eigenpairs of $P(\lambda)$ gain a lot in accuracy when using the QZ algorithm on an appropriate scaling of the pencil (1.2).

*Received by the editors September 17, 2003; accepted for publication (in revised form) by N. J. Higham February 9, 2004; published electronically September 14, 2004. This research was supported by the Belgian Programme on Inter-university Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture.
http://www.siam.org/journals/simax/26-1/43491.html

†Department of Mathematics, National Tsing Hua University, Hsinchu, 300 Taiwan (d887206@am.nthu.edu.tw, wwlin@am.nthu.edu.tw).

‡Department of Mathematical Engineering, Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium (vdooren@csam.ucl.ac.be).

2. Scaling of second order polynomial matrices. In section 3 of [2], the author considers the scaled QEP defined by

$$\hat{P}(\mu)x \equiv (\mu^2\hat{P}_2 + \mu\hat{P}_1 + \hat{P}_0)x = 0 \tag{2.1}$$

with $\mu = \lambda/\alpha$, $\hat{P}_2 = \alpha^2 P_2$, $\hat{P}_1 = \alpha P_1$ and $\hat{P}_0 = P_0$, where α is a scaling factor, and investigates the possibility of using this scaling of the QEP (1.1) to improve the backward error of the solution obtained via the GEP formulation (1.2). The paper [2] does not solve this scaling problem (the Conclusions section mentions it as an open problem) but instead it derives a sufficient condition to verify the backward stability for the QEP. Here we restate this theorem without a proof (see Theorem 7 in [2]).

THEOREM 2.1 (see [2]). *If $\|P_i\|_2 = 1$, $i = 0, 1, 2$, then solving the GEP (1.2) with a backward stable algorithm (e.g., the QZ algorithm) for the GEP is backward stable for the QEP: there exist perturbations Δ_i, $i = 0, 1, 2$ with norms of the order of the machine precision ϵ, such that $[\lambda^2(P_2 + \Delta_2) + \lambda(P_1 + \Delta_1) + (P_0 + \Delta_0)]\xi = 0$ for every computed eigenpair (λ, ξ).*

Remark. Theorem 2.1 is similar to a result given in an earlier paper [4], for a pencil (1.1) with $\|A\|_2 \approx \|B\|_2 \approx 1$. It is shown there that for any perturbations $\|\delta A\|_2 \approx \|\delta B\|_2 \approx \epsilon$ (e.g., the backward errors resulting from the QZ algorithm) there exist transformations $S := I + E$ and $T := I + F$ such that

$$S[\lambda(A + \delta A) - (B + \delta B)]T = \lambda \begin{bmatrix} I & 0 \\ 0 & (P_2 + \Delta_2) \end{bmatrix} - \begin{bmatrix} 0 & I \\ -(P_0 + \Delta_0) & -(P_1 + \Delta_1) \end{bmatrix},$$

where $\Delta_0, \Delta_1, \Delta_2, E$, and F have norms of the order of the machine precision. □

The above results suggest that a good scaling strategy for the QEP (1.1) is to scale P_2, P_1, and P_0 so that their 2-norms are all close to 1. Consider modifying the polynomial matrix $P(\lambda) = \lambda^2 P_2 + \lambda P_1 + P_0$ as follows:

$$\mu\alpha = \lambda; \quad \tilde{P}(\mu) \equiv P(\lambda)\beta = \mu^2\left(P_2\alpha^2\beta\right) + \mu\left(P_1\alpha\beta\right) + \left(P_0\beta\right) \tag{2.2}$$

which yields a corresponding matrix pencil

$$\mu\tilde{B} - \tilde{A} = \mu \begin{bmatrix} I & 0 \\ 0 & \tilde{P}_2 \end{bmatrix} - \begin{bmatrix} 0 & I \\ -\tilde{P}_0 & -\tilde{P}_1 \end{bmatrix} \tag{2.3}$$

with coefficient matrices $\tilde{P}_2 = P_2\alpha^2\beta$, $\tilde{P}_1 = P_1\alpha\beta$, $\tilde{P}_0 = P_0\beta$ of respective 2-norms $\tilde{\gamma}_2 = \gamma_2\alpha^2\beta$, $\tilde{\gamma}_1 = \gamma_1\alpha\beta$, $\tilde{\gamma}_0 = \gamma_0\beta$. One should thus try to minimize the maximum distance

$$\min_{\alpha,\beta} \max\left\{|\beta\alpha^2\gamma_2 - 1|,\ |\beta\alpha\gamma_1 - 1|,\ |\beta\gamma_0 - 1|\right\}. \tag{2.4}$$

If we substitute $\hat{\alpha} := \alpha\sqrt{\gamma_2/\gamma_0}$, $\hat{\beta} := \beta\gamma_0$, and $\hat{\gamma} := \gamma_1/\sqrt{\gamma_2\gamma_0}$, then this reduces to

$$\min_{\hat{\alpha},\hat{\beta}} \max\left\{|\hat{\beta}\hat{\alpha}^2 - 1|,\ |\hat{\beta}\hat{\alpha}\hat{\gamma} - 1|,\ |\hat{\beta} - 1|\right\}.$$

At the optimum, all three quantities will be equal since otherwise we can decrease the maximum by adapting $\hat{\beta}$ and $\hat{\alpha}$. Hence we must have

$$|\hat{\beta}\hat{\alpha}^2 - 1| = |\hat{\beta}\hat{\alpha}\hat{\gamma} - 1| = |\hat{\beta} - 1|.$$

Since at least two of the quantities inside $|\cdot|$ must also have equal signs, one of the following three relations must hold at the optimum:

$$\hat{\alpha}^2 = 1, \quad \text{or} \quad \hat{\alpha}\hat{\gamma} = 1, \quad \text{or} \quad \hat{\alpha} = \hat{\gamma}.$$

By mere comparison, one then finds that the optimum $\hat{\alpha}^*$ is given by the first choice, which finally yields

$$\hat{\alpha}^* = 1, \quad \hat{\beta}^* = 2/(1+\hat{\gamma}).$$

In terms of the original variables we thus have

$$\alpha^* = \sqrt{\gamma_0/\gamma_2}, \quad \beta^* = 2/(\gamma_0 + \gamma_1\sqrt{\gamma_0/\gamma_2})$$

and the new values for the scaled norms are

$$\tilde{\gamma}_0 = \tilde{\gamma}_2 = 2/(1+\hat{\gamma}), \quad \tilde{\gamma}_1 = 2\hat{\gamma}/(1+\hat{\gamma}),$$

while

$$\max\{|\tilde{\gamma}_2 - 1|, |\tilde{\gamma}_1 - 1|, |\tilde{\gamma}_0 - 1|\} = |(1-\hat{\gamma})/(1+\hat{\gamma})|.$$

We point out that bounding (2.4) also implies bounding the normwise backward error of the matrices $\tilde{P}_i$. Indeed, one easily checks that $\|\tilde{A}\|_2 \leq 2$ and $\|\tilde{B}\|_2 \leq \sqrt{5}$. When running the QZ algorithm on $\mu\tilde{B} - \tilde{A}$ we will have—according to the above remark—equivalent absolute backward errors $\tilde{\Delta}_i$, $i = 0, 1, 2$ with norms of the order of the machine precision ϵ. The *structured relative backward errors* will therefore be of the order of

$$\|\tilde{\Delta}_0\|_2/\|\tilde{P}_0\|_2 \approx \|\tilde{\Delta}_2\|_2/\|\tilde{P}_2\|_2 \approx \epsilon(1+\hat{\gamma}), \quad \|\tilde{\Delta}_1\|_2/\|\tilde{P}_1\|_2 \approx \epsilon(1+\hat{\gamma})/\hat{\gamma}, \tag{2.5}$$

and $\max\left\{1+\hat{\gamma}, 1+\hat{\gamma}^{-1}\right\}$ can thus be seen as a *growth factor* between unstructured relative backward errors on the pencil $\mu\hat{B} - \hat{A}$ and structured relative backward errors on the second order polynomial matrix $\hat{P}(\mu) = \mu^2\hat{P}_2 + \mu\hat{P}_1 + \hat{P}_0$. In the numerical examples section we indeed show that the backward error of an approximate eigenpair (ξ, λ) computed with this optimal scaling strategy improves a lot. Moreover, if $\hat{\gamma} = 1$ (this is, when $\gamma_1^2 = \gamma_0\gamma_2$) then the normwise backward error will be of the order of the machine precision according to Theorem 2.1.

Remark. One could consider a more general type of scaling

$$\mu\tilde{B} - \tilde{A} = \begin{bmatrix} \ell_1 I & 0 \\ 0 & \ell_2 I \end{bmatrix} (\alpha\mu B - A) \begin{bmatrix} r_1 I & 0 \\ 0 & r_2 I \end{bmatrix} \tag{2.6}$$

involving 5 parameters, ℓ_1, ℓ_2, r_1, r_2, and α, but this is in fact the same problem. Dividing ℓ_1, ℓ_2 and multiplying r_1, r_2 by a common factor yields the same solution, so we can choose $r_1 = 1$. Moreover, setting one block norm equal to 1 in both $\tilde{B}$ and $\tilde{A}$ does not modify relative block norms in each individual block, so we can set $\alpha\ell_1 r_1 = 1$ and $\ell_1 r_2 = 1$. This then yields the parametrization $\ell_1 = 1/\alpha$, $\ell_2 = \beta$, $r_1 = 1$, $r_2 = \alpha$, which is exactly the problem we studied above.

We point out that in [1] the more general problem of optimal scaling of companion pencils is considered, but the technique and results are quite different. One could also consider other GEPs with the same generalized eigenvalues as (1.2) (see [3]), but the proposed scaling would then probably have to be adapted.

3. Numerical examples. When applying the QZ algorithm to $\mu\tilde{B} - \tilde{A}$, each *computed eigenpair* μ, ξ satisfies $(\mu\tilde{B} - \tilde{A})\xi \approx 0$. Both subvectors $\xi_1 := \xi(1:n)$, $\xi_2 := \xi(n+1:2n)$ should be proportional to each other and will yield $P(\lambda)\xi_i \approx 0$, where $\lambda = \mu\alpha$. The normwise backward errors $\Delta_i, i = 0, 1, 2$, that are *compatible* with the computed eigenpair

$$[\lambda^2(P_2 + \Delta_2) + \lambda(P_1 + \Delta_1) + (P_0 + \Delta_0)]\xi_j = 0$$

can be bounded using the residuals $P(\lambda)\xi_j$. In [3] it is shown that the smallest normwise backward error satisfies

$$\max_{i=0,1,2} \|\Delta_i\|_2/\|P_i\|_2 = \eta(\xi_j, \lambda) \equiv \frac{\|P(\lambda)\xi_j\|}{(|\lambda|^2\|P_2\| + |\lambda|\|P_1\| + \|P_0\|)\|\xi_j\|} \quad \text{for } j = 1, 2.$$

In the following examples, we use these quantities as measure for the backward error for each eigenpair computed by the QZ algorithm. The quantities $\eta_s(\xi_j, \lambda)$, $j = 1, 2$, on the other hand, refer to the computed eigenvector/eigenvalue pairs obtained *after scaling*. All computations were performed using MATLAB/Version 6.0 on a Compaq/DS20 workstation. The machine precision is 1.1×10^{-16}.

Example 1. We first consider the nuclear power plant problem in [3]. The backward errors of the computed eigenpairs corresponding to the smallest and largest eigenvalues in modulus, and the corresponding *scaled* backward errors are shown in Table 3.1. In this example, the 2-norms of the matrices P_2, P_1, P_0 are of the order of 10^8, 10^{10}, and 10^{13}, respectively. After applying the optimal scaling presented in section 2, their 2-norms are reduced to $\tilde{\gamma}_0 = \tilde{\gamma}_2 \approx 1.18$, $\tilde{\gamma}_1 \approx 0.821$, respectively. For this example $\hat{\gamma} = 0.697$, which implies that the scaled backward errors should be of the order of the machine precision.

TABLE 3.1
Backward errors for Example 1.

$\lvert\lambda\rvert$	$\eta(\xi_1, \lambda)$	$\eta(\xi_2, \lambda)$	$\eta_s(\xi_1, \lambda)$	$\eta_s(\xi_2, \lambda)$
17.7	3e-5	6e-8	3e-15	1e-16
361	2e-11	2e-11	1e-18	2e-18

Example 2. Here we tested randomly generated second order polynomial matrices $P(\lambda)$ with $\|P_2\|_2 = O(10^5)$, $\|P_1\|_2 = O(10^3)$, $\|P_0\|_2 = O(10^{-3})$, and $n = 10$, respectively. The absolute values of computed eigenvalues range between $O(10^{-2})$ and $O(10^{-7})$ and in Table 3.2 we give the backward errors of the 5 eigenpairs of smallest modulus, computed without and with scaling. With the optimal scaling, the 2-norms of the scaled coefficient matrices $\tilde{P}_2$, $\tilde{P}_1$, and $\tilde{P}_0$ are reduced to $\tilde{\gamma}_0 = \tilde{\gamma}_2 \approx 2.13 \times 10^{-2}$, $\tilde{\gamma}_1 \approx 1.98$, respectively. For this example $\hat{\gamma} = 93.01$, which means that after scaling we should not lose more than one or two digits of accuracy, which is confirmed in the experiments.

Example 3. In this example we tested randomly generated second order polynomial matrices $P(\lambda)$ with $\|P_2\|_2 \approx 5.54 \times 10^{-5}$, $\|P_1\|_2 \approx 4.73 \times 10^3$, $\|P_0\|_2 \approx 6.01 \times 10^{-3}$, and $n = 10$, respectively. The absolute values of computed eigenvalues range between $O(10^{-7})$ and $O(10^8)$. In Table 3.3 we give the backward errors of the 5 eigenpairs of smallest modulus without and with scaling. The scaled 2-norms are reduced to $\tilde{\gamma}_0 = \tilde{\gamma}_2 \approx 2.44 \times 10^{-7}$, $\tilde{\gamma}_1 \approx 2.00$, respectively, and $\hat{\gamma} \approx 8.19 \times 10^6$. This implies that after scaling we should not lose more than six digits of accuracy.

Example 4. Here we also tested randomly generated second order polynomial matrices $P(\lambda)$ in (1.1) with $\|P_2\|_2 \approx 5.03 \times 10^5$, $\|P_1\|_2 \approx 6.53 \times 10^{-3}$, $\|P_0\|_2 \approx$

TABLE 3.2
Backward errors for Example 2.

$\|\lambda\|$	$\eta(\xi_1,\lambda)$	$\eta(\xi_2,\lambda)$	$\eta_s(\xi_1,\lambda)$	$\eta_s(\xi_2,\lambda)$
2.40e-7	5e-8	4e-7	5e-16	3e-15
4.04e-7	6e-8	3e-7	1e-15	3e-15
6.47e-7	3e-8	8e-8	4e-16	2e-15
6.70e-7	2e-8	6e-8	9e-16	3e-15
1.22e-6	5e-9	7e-9	3e-16	2e-15

TABLE 3.3
Backward errors for Example 3.

$\|\lambda\|$	$\eta(\xi_1,\lambda)$	$\eta(\xi_2,\lambda)$	$\eta_s(\xi_1,\lambda)$	$\eta_s(\xi_2,\lambda)$
2.09e-7	2e-7	1e-6	6e-11	2e-10
5.71e-7	2e-7	5e-7	2e-10	2e-10
7.44e-7	2e-7	6e-7	3e-11	3e-11
1.37e-6	2e-7	1e-7	3e-11	2e-11
1.62e-6	2e-7	1e-7	7e-12	5e-12

6.06×10^3, and $n = 10$, respectively. The absolute values of computed eigenvalues range between $O(10^{-2})$ and $O(10^{-1})$. In Table 3.4 we give the backward errors of the 5 eigenpairs of smallest modulus without and with scaling. The scaled 2-norms are now $\tilde{\gamma}_0 = \tilde{\gamma}_2 \approx 2.00$, $\tilde{\gamma}_1 \approx 2.37 \times 10^{-7}$, respectively, and the backward errors of the computed eigenpairs are reported in Table 3.4. In this case, $\hat{\gamma} \approx 1.18 \times 10^{-7}$ which means that after scaling we should not lose more than six digits of accuracy.

TABLE 3.4
Backward errors for Example 4.

$\|\lambda\|$	$\eta(\xi_1,\lambda)$	$\eta(\xi_2,\lambda)$	$\eta_s(\xi_1,\lambda)$	$\eta_s(\xi_2,\lambda)$
1.72e-2	2e-13	1e-11	7e-16	3e-15
7.21e-2	1e-12	6e-12	5e-16	6e-16
1.06e-1	1e-12	5e-12	6e-16	6e-16
1.13e-1	1e-12	2e-12	3e-16	5e-16
1.55e-1	2e-12	2e-12	5e-16	6e-16

As shown in all of these examples, the backward errors are all significantly improved by the scaling: we gain up to 10 digits of accuracy! Also the computable quantity $\hat{\gamma}$ gives an upper bound on the backward error which is often a good estimate as well, except for the last example where the accuracy is much better than predicted.

REFERENCES

[1] D. LEMONNIER AND P. VAN DOOREN, *Optimal scaling of companion pencils for the QZ algorithm*, Proceedings SIAM Applied Linear Algebra Conference, Williamsburg, July 2003, http://www.siam.org/meetings/la03/proceedings/lemonnid.pdf.

[2] F. TISSEUR, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra Appl., 309 (2000), pp. 339–361.

[3] F. TISSEUR AND K. MEERBERGEN, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.

[4] P. VAN DOOREN AND P. DEWILDE, *The eigenstructure of an arbitrary polynomial matrix: Computational aspects*, Linear Algebra Appl., 50 (1983), pp. 545–579.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 1, pp. 257–271

MINIMAL SPECTRALLY ARBITRARY SIGN PATTERNS*

T. BRITZ†, J. J. MCDONALD‡, D. D. OLESKY§, AND P. VAN DEN DRIESSCHE†

Abstract. An $n \times n$ sign pattern $\mathcal{A}$ is spectrally arbitrary if given any self-conjugate spectrum there exists a matrix realization of $\mathcal{A}$ with that spectrum. If replacing any nonzero entry of $\mathcal{A}$ by zero destroys this property, then $\mathcal{A}$ is a minimal spectrally arbitrary sign pattern. Several families of sign patterns are presented that, for all $n \geq 3$, each contain an $n \times n$ minimal spectrally arbitrary sign pattern. These are the first families proven to have this property, and they improve previously known results. Furthermore, all 3×3 minimal spectrally arbitrary sign patterns are determined, it is proved that any irreducible $n \times n$ spectrally arbitrary sign pattern must have at least $2n-1$ nonzero entries, and it is conjectured that the minimum number of nonzero entries is $2n$.

Key words. spectrum, sign pattern, nilpotent matrix

AMS subject classifications. 15A18, 15A48

DOI. 10.1137/S0895479803432514

1. Introduction.

A sign pattern is a square matrix with entries in $\{+, -, 0\}$. If $\mathcal{A}$ is a sign pattern and A is a real matrix for which each entry has the same sign as the corresponding entry of $\mathcal{A}$, then A is said to be a *realization* of $\mathcal{A}$, and we write $A \in \mathcal{A}$. This convention is also used for zero-nonzero patterns $\mathcal{A}$. A sign pattern $\mathcal{B} = [b_{ij}]$ is a *superpattern* of a sign pattern $\mathcal{A} = [a_{ij}]$ if $b_{ij} = a_{ij}$ whenever $a_{ij} \neq 0$. Similarly, $\mathcal{B}$ is a *subpattern* of $\mathcal{A}$ if $b_{ij} = 0$ whenever $a_{ij} = 0$. Note that each sign pattern is a superpattern and a subpattern of itself. An $n \times n$ sign pattern $\mathcal{A}$ is *spectrally arbitrary* if for each real monic polynomial $r(x)$ of degree n, there exists some $A \in \mathcal{A}$ with characteristic polynomial $p_A(x) = r(x)$. Thus, $\mathcal{A}$ is spectrally arbitrary if, given any self-conjugate spectrum, there exists $A \in \mathcal{A}$ with that spectrum. A sign pattern $\mathcal{A}$ is *minimally spectrally arbitrary* if it is spectrally arbitrary but is not spectrally arbitrary if any nonzero entry of $\mathcal{A}$ is replaced by zero. If $\mathcal{A}$ is an $n \times n$ sign pattern or zero-nonzero pattern, then $\mathcal{A}$ *allows nilpotency* if there exists some $A \in \mathcal{A}$ with characteristic polynomial $p_A(x) = x^n$. Note that each spectrally arbitrary sign pattern must allow nilpotency, must be inertially arbitrary (as explained below Theorem 2.5), and must also be potentially stable. These are three important sign pattern problems that are considered in the literature (see, for example, [1, 3, 4, 5, 7, 8, 9]).

In [8, Theorem 2.6], it is proved that a *p-striped sign pattern*—that is, an $n \times n$ ($n \geq 2$) sign pattern having p ($1 \leq p \leq n-1$) columns all of whose entries are positive and $n-p$ columns all of whose entries are negative—is spectrally arbitrary. The proof is based on constructions using a Soules matrix, and gives (as far as we are aware) the first spectrally arbitrary sign pattern for all $n \geq 2$.

*Received by the editors July 29, 2003; accepted for publication (in revised form) by R. Bhatia February 18, 2004, published electronically September 14, 2004. This work was supported through research grants from the Natural Sciences and Engineering Research Council of Canada. The work of the first author was also supported by a PIMS Postdoctoral Fellowship.

http://www.siam.org/journals/simax/26-1/43251.html

†Department of Mathematics and Statistics, University of Victoria, Victoria, BC V8W 3P4, Canada (britz@math.uvic.ca, pvdd@math.uvic.ca).

‡Mathematics Department, Washington State University, Pullman, WA 99164-3113 (jmcdonald@math.wsu.edu).

§Department of Computer Science, University of Victoria, Victoria, BC V8W 3P6, Canada (dolesky@cs.uvic.ca).

Each p-striped sign pattern is full, and current interest is in determining minimal spectrally arbitrary patterns. In section 2, an $n \times n$ ($n \geq 3$) irreducible sign pattern $\mathcal{V}_n$ is presented and proved to be minimally spectrally arbitrary. To our knowledge, no such family of minimal spectrally arbitrary sign patterns has been presented previously. Each of these sign patterns is a Hessenberg matrix and all superpatterns of these sign patterns are shown to be spectrally arbitrary. This strengthens results in [5].

In section 3, the family of sign patterns $\mathcal{V}_n$ is extended to a larger family of $n \times n$ irreducible sign patterns $\mathcal{W}_n(k)$ with each superpattern shown to be spectrally arbitrary. This provides an alternate proof that every p-striped pattern is spectrally arbitrary [8]. The sign pattern $\mathcal{W}_n(k)$ is not necessarily minimally spectrally arbitrary. However, the minimal spectrally arbitrary sign patterns that are contained in $\mathcal{W}_n(k)$ are characterized.

The family of sign patterns $\mathcal{V}_n$ is generalized in another way in section 4 by introducing a family of zero-nonzero patterns $\mathcal{V}_n^*(I)$. It is shown that if $\mathcal{V}_n^*(I)$ allows nilpotency, then $\mathcal{V}_n^*(I)$ determines an $n \times n$ irreducible sign pattern $\mathcal{V}_n(I)$ that is minimally spectrally arbitrary with each superpattern being spectrally arbitrary. Two families of irreducible minimal spectrally arbitrary patterns that arise in this manner are described.

Two sign patterns $\mathcal{A}$ and $\mathcal{B}$ are *equivalent* if $\mathcal{B}$ may be obtained from $\mathcal{A}$ by some combination of negation, transposition, permutation similarity, and signature similarity. Note that if $\mathcal{A}$ and $\mathcal{B}$ are equivalent, then $\mathcal{A}$ is spectrally arbitrary if and only if $\mathcal{B}$ is spectrally arbitrary. In section 5, the family of spectrally arbitrary 3×3 sign patterns is characterized explicitly (up to equivalence).

In the concluding section 6, it is proved that any $n \times n$ irreducible spectrally arbitrary sign pattern must contain at least $2n - 1$ nonzero entries. It is conjectured that it must in fact contain at least $2n$ nonzero entries.

2. Hessenberg sign patterns $\mathcal{V}_n$. Results throughout rely heavily upon the following lemma, which is stated as Observations 10 and 15 in [1] and is proved using the implicit function theorem. Let $x_1, \ldots, x_n$ be real variables, and for each $i = 1, \ldots, n$, let $\alpha_i = \alpha_i(x_1, \ldots, x_n)$ be a real function of $(x_1, \ldots, x_n)$ that is continuous and differentiable in each x_j. The Jacobian $J = \frac{\partial(\alpha_1, \ldots, \alpha_n)}{\partial(x_1, \ldots, x_n)}$ is the $n \times n$ matrix with (i, j) entry equal to $\frac{\partial \alpha_i}{\partial x_j}$ for $1 \leq i, j \leq n$.

LEMMA 2.1 (see [1]). *Let $\mathcal{A}$ be an $n \times n$ sign pattern, and suppose that there exists some nilpotent $A \in \mathcal{A}$ with at least n nonzero entries, say $a_{i_1 j_1}, \ldots, a_{i_n j_n}$. Let X be the matrix obtained by replacing these entries in A by variables $x_1, \ldots, x_n$, and let*

$$p_X(x) = x^n - \alpha_1 x^{n-1} + \alpha_2 x^{n-2} - \cdots + (-1)^{n-1} \alpha_{n-1} x + (-1)^n \alpha_n.$$

If $J = \frac{\partial(\alpha_1, \ldots, \alpha_n)}{\partial(x_1, \ldots, x_n)}$ is nonsingular at $(x_1, \ldots, x_n) = (a_{i_1 j_1}, \ldots, a_{i_n j_n})$, then every superpattern of $\mathcal{A}$ is spectrally arbitrary.

Example 2.2. Let $\mathcal{A} = \begin{bmatrix} + & - \\ + & - \end{bmatrix}$. Then $A = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} \in \mathcal{A}$ is nilpotent. Let $X = \begin{bmatrix} x_1 & -1 \\ 1 & x_2 \end{bmatrix}$. Then

$$p_X(x) = x^2 - \alpha_1 x + \alpha_2,$$

where $\alpha_1 = x_1 + x_2$ and $\alpha_2 = x_1x_2 + 1$. Thus

$$J = \frac{\partial(\alpha_1, \alpha_2)}{\partial(x_1, x_2)} = \begin{bmatrix} 1 & 1 \\ x_2 & x_1 \end{bmatrix} \quad \text{and} \quad \det J = x_1 - x_2.$$

At $(x_1, x_2) = (1, -1)$, $\det J = 2 \neq 0$. By Lemma 2.1, $\mathcal{A}$ is spectrally arbitrary, and it is easily seen that it is minimal. Note that up to equivalence, $\mathcal{A}$ is the unique (minimal) spectrally arbitrary 2×2 sign pattern.

Given a sign pattern $\mathcal{A}$, let $D(\mathcal{A})$ be its associated digraph. For any digraph D, let $G(D)$ denote the underlying multigraph of D, i.e., the graph obtained from D by ignoring the direction of each arc. The following lemma is well known and can be proved by induction. We use this to normalize an $n \times n$ matrix $A \in \mathcal{A}$ by fixing up to $n - 1$ entries to have magnitude 1.

LEMMA 2.3. *Let $\mathcal{A}$ be an $n \times n$ sign pattern and let $A \in \mathcal{A}$. If T is a subdigraph of $D(\mathcal{A})$ such that $G(T)$ is a forest, then $\mathcal{A}$ has a realization that is positive diagonally similar to A such that each entry corresponding to an arc of T has magnitude 1. In particular, if $\mathcal{A}$ is irreducible, then $G(D(\mathcal{A}))$ contains a spanning tree, and $\mathcal{A}$ must therefore have a realization with at least $n - 1$ off-diagonal entries in $\{-1, 1\}$ that is positive diagonally similar to A.*

Let $n \geq 3$, and consider the $n \times n$ Hessenberg sign pattern

$$\mathcal{V}_n = \begin{bmatrix} + & - & 0 & 0 & 0 & 0 \\ + & 0 & - & 0 & 0 & 0 \\ \vdots & 0 & 0 & \ddots & 0 & 0 \\ + & 0 & 0 & 0 & - & 0 \\ + & 0 & 0 & 0 & 0 & - \\ + & 0 & 0 & 0 & 0 & - \end{bmatrix}.$$

THEOREM 2.4. *For $n \geq 3$, the pattern $\mathcal{V}_n$ is a minimal spectrally arbitrary pattern.*

Proof. Let

$$r(x) = x^n - r_1x^{n-1} + r_2x^{n-2} - \cdots + (-1)^{n-1}r_{n-1}x + (-1)^n r_n$$

be a fixed but arbitrary real monic polynomial of degree n. Let

$$A = \begin{bmatrix} a_1 & -1 & 0 & 0 & 0 & 0 \\ a_2 & 0 & -1 & 0 & 0 & 0 \\ \vdots & 0 & 0 & \ddots & 0 & 0 \\ a_{n-2} & 0 & 0 & 0 & -1 & 0 \\ a_{n-1} & 0 & 0 & 0 & 0 & -1 \\ a_n & 0 & 0 & 0 & 0 & -t \end{bmatrix}.$$

The characteristic polynomial of A is

$$p_A(x) = x^n - \alpha_1x^{n-1} + \alpha_2x^{n-2} - \cdots + (-1)^{n-1}\alpha_{n-1}x + (-1)^n\alpha_n,$$

where $\alpha_1 = a_1 - t$, and $\alpha_i = a_i - ta_{i-1}$ for $i = 2, \ldots, n$. Set $a_1 = r_1 + t$. For each $i = 2, \ldots, n$, set

$$a_i = t^i + \sum_{j=1}^{i} r_j t^{i-j}.$$

Then $\alpha_1 = a_1 - t = r_1 + t - t = r_1$, and for $i = 2, \ldots, n$,

$$\alpha_i = a_i - ta_{i-1} = \left(t^i + \sum_{j=1}^{i} r_j t^{i-j} \right) - t \left(t^{i-1} + \sum_{j=1}^{i-1} r_j t^{i-1-j} \right) = r_i.$$

Thus, $\alpha_i = r_i$ for all $i = 1, \ldots, n$, i.e., $p_A(x) = r(x)$. For all $t > 0$ sufficiently large, each $a_j > 0$ $(1 \leq j \leq n)$ and thus $A \in \mathcal{V}_n$. Hence, $\mathcal{V}_n$ is spectrally arbitrary.

By Lemma 2.3, each matrix with sign pattern $\mathcal{V}_n$ is positive diagonally similar to a matrix A in the above form. If one of the -1 entries in columns $2, \ldots, n-1$ of A is replaced by zero, then the resulting matrix is necessarily singular. Similarly, if $t = 0$ or the -1 entry in column n of A is replaced by zero, then the resulting matrix is necessarily nonsingular. If $a_i = 0$ for some $1 \leq i \leq n$, then $\alpha_i \leq 0$. Thus, $\mathcal{V}_n$ is minimally spectrally arbitrary. □

Set $t = 1$ in the matrix A from the above proof. If $a_1 = \cdots = a_n = 1$, then A is nilpotent. The Jacobian $J = \frac{\partial(\alpha_1, \ldots, \alpha_n)}{\partial(a_1, \ldots, a_n)}$ has 1 in each diagonal position, $-t = -1$ in each subdiagonal position, and zeros elsewhere. Thus, $\det J = 1 \neq 0$. Hence, the theorem below follows from Lemma 2.1.

THEOREM 2.5. *For $n \geq 3$, any superpattern of $\mathcal{V}_n$ is a spectrally arbitrary pattern.*

An $n \times n$ sign pattern $\mathcal{A}$ is *inertially arbitrary* if given a nonnegative triple of integers (n_1, n_2, n_3) with $n_1 + n_2 + n_3 = n$, there exists some $A \in \mathcal{A}$ that has n_1 eigenvalues with positive real part, n_2 eigenvalues with negative real part, and n_3 eigenvalues with zero real part. Note that if a sign pattern is spectrally arbitrary, then it is also inertially arbitrary. Recently, several families of sign patterns have been shown to be inertially arbitrary (see [5, 8, 9]). The sign patterns described in [5] are superpatterns of the pattern $\mathcal{V}_n$. It follows from Theorem 2.5 that these sign patterns are not only inertially arbitrary but indeed spectrally arbitrary.

3. Non-Hessenberg sign patterns $\mathcal{W}_n(k)$. We now define a general class of $n \times n$ sign patterns that includes the Hessenberg patterns $\mathcal{V}_n$. Let $n \geq 3$ and $0 \leq k \leq n-2$ be given. Define $\mathcal{W}_n(k)$ to be the $n \times n$ sign pattern with positive signs throughout the first column and in the entries

$$\{(j, j+1) \,:\, j = 1, \ldots, k\};$$

negative signs in the entries

$$\{(j, j+1) \,:\, j = k+1, \ldots, n-1\},\ \{(j, n) \,:\, j = 1, \ldots, k\}, \quad \text{and} \quad (n, n);$$

and zeros elsewhere. For $k \geq 1$, let $W_n(k) \in \mathcal{W}_n(k)$ have values $a_1, \ldots, a_n$ in column 1; $-b_1, \ldots, -b_k$ in the first k entries of column n; $-b_n$ in the (n, n) entry; and all entries on the superdiagonal have magnitude 1. For example, the sign pattern $\mathcal{W}_7(3)$ and a realization $W_7(3)$ are

$$\begin{bmatrix} + & + & 0 & 0 & 0 & 0 & - \\ + & 0 & + & 0 & 0 & 0 & - \\ + & 0 & 0 & + & 0 & 0 & - \\ + & 0 & 0 & 0 & - & 0 & 0 \\ + & 0 & 0 & 0 & 0 & - & 0 \\ + & 0 & 0 & 0 & 0 & 0 & - \\ + & 0 & 0 & 0 & 0 & 0 & - \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} a_1 & 1 & 0 & 0 & 0 & 0 & -b_1 \\ a_2 & 0 & 1 & 0 & 0 & 0 & -b_2 \\ a_3 & 0 & 0 & 1 & 0 & 0 & -b_3 \\ a_4 & 0 & 0 & 0 & -1 & 0 & 0 \\ a_5 & 0 & 0 & 0 & 0 & -1 & 0 \\ a_6 & 0 & 0 & 0 & 0 & 0 & -1 \\ a_7 & 0 & 0 & 0 & 0 & 0 & -b_7 \end{bmatrix},$$

respectively. Then matrix $W_n(k) \in \mathcal{W}_n(k)$ has characteristic polynomial

$$x^n - \alpha_1 x^{n-1} + \alpha_2 x^{n-2} - \cdots + (-1)^{n-1}\alpha_{n-1}x + (-1)^n \alpha_n,$$

where

$$\begin{aligned}
&\alpha_1 = a_1 - b_n, \\
&\alpha_i = (-1)^{i-1}(a_i + a_{i-1}b_n - b_{i-1}a_n) && \text{for } i = 2, \ldots, k+1, \\
&\alpha_i = (-1)^{k}(a_i - a_{i-1}b_n) && \text{for } i = k+2, \ldots, n.
\end{aligned}$$

PROPOSITION 3.1. *For each pair $n \geq 3$ and $0 \leq k \leq n-2$, the pattern $\mathcal{W}_n(k)$ is a spectrally arbitrary pattern, and any superpattern of $\mathcal{W}_n(k)$ is spectrally arbitrary.*

Proof. Since the patterns $\mathcal{W}_n(0)$ are the Hessenberg patterns $\mathcal{V}_n$, the result for $k = 0$ follows from Theorem 2.5.

Let $1 \leq k \leq n-2$ be given. Note that $W_n(k)$ is nilpotent if $a_1 = \cdots = a_n = b_n = 1$ and $b_1 = \cdots = b_k = 2$. Now set $b_1 = \cdots = b_k = 2$ and $b_n = 1$, leaving $a_1, \ldots, a_n$ as variables. Then the terms of the characteristic polynomial of $W_n(k)$ are

$$\begin{aligned}
&\alpha_1 = a_1 - 1, \\
&\alpha_i = (-1)^{i-1}(a_i + a_{i-1} - 2a_n) && \text{for } i = 2, \ldots, k+1, \\
&\alpha_i = (-1)^{k}(a_i - a_{i-1}) && \text{for } i = k+2, \ldots, n.
\end{aligned}$$

The Jacobian $J = \frac{\partial(\alpha_1, \ldots, \alpha_n)}{\partial(a_1, \ldots, a_n)}$ is a matrix with ± 1 entries on the main diagonal and on the subdiagonal, and (i, n) entries equal to $(-1)^i 2$ for $i = 2, \ldots, k+1$. Thus, J has determinant of the form $\pm 1 + 2c$ for some integral constant c, and the result follows from Lemma 2.1. ∎

COROLLARY 3.2 (see [8, Theorem 2.6]). *For $n \geq 2$, every $n \times n$ p-striped sign pattern is spectrally arbitrary.*

Proof. The case $n = 2$ is proved in Example 2.2. Suppose that $n \geq 3$, and consider the $n \times n$ p-striped sign pattern with precisely $p = k + 1 \leq n - 1$ positive columns for some $k \geq 0$. By permutation similarity, it may be assumed that the first $k + 1$ columns are positive. This p-striped sign pattern is a superpattern of $\mathcal{W}_n(k)$, and the result follows by Proposition 3.1. ∎

If $k = 0$, then $\mathcal{W}_n(0) = \mathcal{V}_n$ is a minimal spectrally arbitrary pattern. For $k = 1$, $\mathcal{W}_n(1)$ is minimally spectrally arbitrary, since at least one of the coefficients α_i has fixed sign if any of the variables $a_1, \ldots, a_n, b_1, b_n$ are set to zero. This is not necessarily true for values $k \geq 2$. For such k, let $\mathcal{I}_k$ denote the family of subsets $I \subseteq \{2, \ldots, k\}$ such that I does not contain two consecutive integers $i, i+1$, and $\{1, \ldots, k+1\} \backslash I$ does not contain three consecutive integers $i, i+1, i+2$. Note that the set of all even integers and the set of all odd integers in $\{2, \ldots, k\}$ both are members of $\mathcal{I}_k$. For $I \in \mathcal{I}_k$, set $a_i = 0$ for each $i \in I$, and let the resulting sign pattern and matrix be denoted by $\mathcal{W}_n^I(k)$ and $W_n^I(k)$, respectively.

THEOREM 3.3. *For each pair $n \geq 4$ and $2 \leq k \leq n-2$, the family of minimal spectrally arbitrary subpatterns of $\mathcal{W}_n(k)$ consists of the patterns $\mathcal{W}_n^I(k)$, where $I \in \mathcal{I}_k$. Furthermore, any superpattern of these patterns is spectrally arbitrary.*

Proof. Let $I \in \mathcal{I}_k$, and set $a_n = b_n = 1$ in $W_n^I(k)$. Then $W_n^I(k)$ is nilpotent if and

only if the following coefficients all equal 0:

$$\begin{aligned}
&\alpha_1 = a_1 - 1,\\
&\alpha_i = (-1)^{i-1}(a_{i-1} - b_{i-1}) && \text{for } i \in I,\\
&\alpha_i = (-1)^{i-1}(a_i - b_{i-1}) && \text{for } i-1 \in I,\ i \in \{2,\dots,k+1\}\backslash I,\\
&\alpha_i = (-1)^{i-1}(a_i + a_{i-1} - b_{i-1}) && \text{for } i-1, i \in \{1,\dots,k+1\}\backslash I,\\
&\alpha_i = (-1)^{k}(a_i - a_{i-1}) && \text{for } i \in \{k+2,\dots,n\}.
\end{aligned}$$

Note that $W_n^I(k)$ is nilpotent if $a_i = 1$ for all variables a_i appearing in the equations above, and for each $i \in \{2,\dots,k+1\}$, the variables $b_{i-1} = 2$ if both i and $i+1$ are contained in $\{1,\dots,k+1\}\backslash I$, and $b_{i-1} = 1$ otherwise. The Jacobian $J = \frac{\partial(\alpha_1,\dots,\alpha_n)}{\partial(a_1,b_1,\dots,b_k,a_{k+1},\dots,a_{n-1})}$ is the direct sum of a lower-triangular $k \times k$ matrix and an upper-triangular $(n-k)\times(n-k)$ matrix, with ± 1 entries on the main diagonal. The determinant of J has magnitude 1, so J is nonsingular. By Lemma 2.1, $\mathcal{W}_n^I(k)$ is spectrally arbitrary, and each superpattern of $\mathcal{W}_n^I(k)$ is also spectrally arbitrary. By the definition of $\mathcal{I}_k$, if any variable a_i, where $i \in \{2,\dots,k\}\backslash I$, is set to 0, then either a_{i-1} or a_{i+1} also equals 0, and the sign of α_i or α_{i+1} is fixed. Thus, $\mathcal{W}_n^I(k)$ is a minimal spectrally arbitrary sign pattern.

Suppose that $\mathcal{W}$ is a minimal spectrally arbitrary subpattern of $\mathcal{W}_n(k)$ with realization W obtained by setting some of the variables spectrally arbitrary, no coefficient α_i has fixed sign. Thus, none of the variables $a_1, a_{k+1}, \dots, a_n, b_1, \dots, b_k, b_n$ equals 0. Furthermore, no two consecutive variables a_{i-1} and a_i can both equal zero. Suppose that $i, i+1, i+2$ are three consecutive integers contained in $\{1,\dots,k+1\}$ such that $a_i, a_{i+1}, a_{i+2} \neq 0$. If the entry a_{i+1} is replaced by a zero, then the resulting sign pattern is also spectrally arbitrary, contradicting the minimality of $\mathcal{W}$. It follows that $\mathcal{W} = \mathcal{W}_n^I(k)$, where $I = \{i :\ 2 \le i \le k, a_i = 0\}$. ∎

4. Sign patterns $\mathcal{V}_n(I)$. For $n \ge 3$, consider the matrix

$$A = \begin{bmatrix}
a_0 & -1 & 0 & 0 & 0 & 0\\
a_1 & 0 & -1 & 0 & 0 & 0\\
a_2 & 0 & 0 & -1 & 0 & 0\\
\vdots & 0 & 0 & 0 & -1 & 0\\
a_{n-2} & 0 & 0 & 0 & 0 & -1\\
a_{n-1} & b_{n-2} & b_{n-3} & \cdots & b_1 & b_0
\end{bmatrix}, \tag{4.1}$$

where the entries a_0, b_0, and a_{n-1} are nonzero, and precisely one of a_i and b_i for each $i = 1,\dots,n-2$ is nonzero. The zero-nonzero pattern determined by A is denoted by $\mathcal{V}_n^*(I)$, where $I = \{i :\ a_i = 0\}$. The matrix A has characteristic polynomial

$$p_A(x) = x^n - \alpha_0 x^{n-1} + \alpha_1 x^{n-2} - \cdots + (-1)^{n-1}\alpha_{n-2}x + (-1)^n\alpha_{n-1},$$

where

$$\begin{aligned}
&\alpha_0 = a_0 + b_0,\\
&\alpha_i = a_i + b_i + \sum_{j=0}^{i-1} a_j b_{i-1-j} \quad \text{for} \quad i = 1,\dots,n-2,\\
\text{and} \quad &\alpha_{n-1} = a_{n-1} + \sum_{j=0}^{n-2} a_j b_{n-2-j}.
\end{aligned}$$

Define $s_i = a_i + b_i$ for $i = 0, \cdots, n-2$ and $s_{n-1} = a_{n-1}$. Since $\frac{\partial \alpha_i}{\partial s_j}$ is zero whenever $j > i$, the Jacobian $J = \frac{\partial(\alpha_0, \ldots, \alpha_{n-1})}{\partial(s_0, \ldots, s_{n-1})}$ is lower triangular. The diagonal entries $\frac{\partial \alpha_i}{\partial s_i}$ each equal 1, so the Jacobian has determinant 1 and is therefore nonsingular.

For nilpotency to hold, each coefficient α_i for $i = 0, \ldots, n-1$ must vanish, i.e.,

$$
\begin{aligned}
0 &= a_0 + b_0, \\
0 &= a_1 + b_1 + a_0 b_0, \\
0 &= a_2 + b_2 + a_0 b_1 + a_1 b_0, \\
&\vdots \\
0 &= a_{n-2} + b_{n-2} + a_0 b_{n-3} + a_1 b_{n-4} + \cdots + a_{n-3} b_0, \\
0 &= a_{n-1} + a_0 b_{n-2} + a_1 b_{n-3} + \cdots + a_{n-2} b_0.
\end{aligned} \tag{4.2}
$$

An induction argument using these equations shows that any nilpotent $A \in \mathcal{V}_n^*(I)$, parameterized as in (4.1), satisfies $a_i = c_i t^{i+1}$ for $i = 0, \ldots, n-1$ and $b_i = d_i t^{i+1}$ for $i = 0, \ldots, n-2$ for some constants $c_0, \ldots, c_{n-1}, d_0, \ldots, d_{n-2}, t$. If c_0 and t are positive, then the sign of a_i for $i = 0, \ldots, n-1$ and the sign of b_i for $i = 0, \ldots, n-2$ are uniquely determined. Thus, if $\mathcal{V}_n^*(I)$ allows nilpotency, then this determines uniquely a sign pattern with a positive $(1,1)$ entry and negative superdiagonal, denoted $\mathcal{V}_n(I)$, which allows nilpotency. By Lemma 2.1, $\mathcal{V}_n(I)$ is a spectrally arbitrary pattern, and each superpattern of $\mathcal{V}_n(I)$ is spectrally arbitrary. It is not difficult to show that $\mathcal{V}_n(I)$ is an irreducible minimal spectrally arbitrary pattern. The preceding discussion gives the following result.

LEMMA 4.1. *If $\mathcal{V}_n^*(I)$ allows nilpotency, then $\mathcal{V}_n(I)$ exists and is minimally spectrally arbitrary, and each superpattern of $\mathcal{V}_n(I)$ is spectrally arbitrary.*

Note that $\mathcal{V}_n^*(\phi)$ allows nilpotency (let $a_0 = \cdots = a_{n-1} = 1$ and $b_0 = -1$) and that $\mathcal{V}_n(\phi) = \mathcal{V}_n$.

LEMMA 4.2. *For $I \subseteq \{1, \ldots, n-2\}$, let $I^C = \{1, \ldots, n-2\} \backslash I$. Then $\mathcal{V}_n^*(I)$ allows nilpotency if and only if $\mathcal{V}_n^*(I^C)$ allows nilpotency. Also, if $\mathcal{V}_n^*(I)$ allows nilpotency, then $\mathcal{V}_{n'}^*(I')$ allows nilpotency for all $3 \leq n' \leq n$, where $I' = \{i \in I : i \leq n' - 2\}$.*

Proof. Note that $\mathcal{V}_n^*(I)$ and $\mathcal{V}_n^*(I^C)$ are equivalent by transposition and permutation similarity. This proves the first statement of the lemma. If $\mathcal{V}_n^*(I)$ allows nilpotency, then equations (4.2) are satisfied by some $A \in \mathcal{V}_n^*(I)$. In particular, the first n' equations are satisfied, so $\mathcal{V}_{n'}^*(I')$ also allows nilpotency. ∎

There are a large number of spectrally arbitrary patterns arising from patterns $\mathcal{V}_n^*(I)$ but they do not generally seem to fall into easily described categories. Numerical evidence suggests that for $n \geq 4$, precisely $2^{n-3} + 2$ of the 2^{n-2} patterns $\mathcal{V}_n^*(I)$ allow nilpotency. The following theorems with $I = \{k\}$ and $I = \{i : 1 \leq i \leq n-2 \text{ is odd}\}$, respectively, describe two classes, $\mathcal{V}_n(k) = \mathcal{V}_n(\{k\})$ and $\mathcal{V}_n^{\text{alt}}$, of minimal spectrally arbitrary sign patterns arising from $\mathcal{V}_n^*(I)$.

Let $n \geq 3$ and $1 \leq k \leq n-2$ be given and define $\mathcal{V}_{n,k}$ to be the $n \times n$ sign pattern with negative signs in the entries

$$\{(j, j+1) : j = 1, \ldots, n-1\},\ \{(j,1) : j = k+2, \ldots, n\},\ \text{and}\ (n,n);$$

positive signs in the entries

$$\{(j,1) : j = 1, \ldots, k\} \quad \text{and} \quad (n, n-k);$$

and zeros elsewhere. Note that $\mathcal{V}_{n,k}$ has the same zero-nonzero pattern as $\mathcal{V}_n^*(k)$. To illustrate,

$$\mathcal{V}_{5,1} = \begin{bmatrix} + & - & 0 & 0 & 0 \\ 0 & 0 & - & 0 & 0 \\ - & 0 & 0 & - & 0 \\ - & 0 & 0 & 0 & - \\ - & 0 & 0 & + & - \end{bmatrix} \quad \text{and} \quad \mathcal{V}_{5,2} = \begin{bmatrix} + & - & 0 & 0 & 0 \\ + & 0 & - & 0 & 0 \\ 0 & 0 & 0 & - & 0 \\ - & 0 & 0 & 0 & - \\ - & 0 & + & 0 & - \end{bmatrix}.$$

THEOREM 4.3. *Let $k \geq 1$ and $k+2 \leq n < 2k + \frac{1}{2}(\sqrt{1+8k}+3)$ be given. Then $\mathcal{V}_n(k)$ exists and is identical to $\mathcal{V}_{n,k}$. Furthermore, it is a minimal spectrally arbitrary pattern, and any superpattern of $\mathcal{V}_n(k)$ is spectrally arbitrary.*

Proof. Let A be as in (4.1), and set

$$a_i = \begin{cases} 1 & \text{for } i = 0, \ldots, k-1, \\ k-i & \text{for } i = k, \ldots, 2k, \\ \frac{1}{2}(i^2 - i) + 2(k^2 - ik) & \text{for } i = 2k+1, \ldots, n-1 \end{cases}$$

and

$$b_i = \begin{cases} -1 & \text{for } i = 0, \\ 1 & \text{for } i = k, \\ 0 & \text{for otherwise}. \end{cases}$$

The polynomial $\frac{1}{2}(x^2 - x) + 2(k^2 - xk)$ has roots $2k + \frac{1}{2} \pm \frac{1}{2}\sqrt{1+8k}$. Thus, the inequality

$$n - 1 < 2k + \frac{1}{2} + \frac{1}{2}\sqrt{1+8k}$$

implies that $\frac{1}{2}(i^2 - i) + 2(k^2 - ik) < 0$ for all $2k+1 \leq i \leq n-1$. Hence, $A \in \mathcal{V}_n^*(k)$ and $A \in \mathcal{V}_{n,k}$. To prove Theorem 4.3, it suffices, by Lemma 4.1, to show that A is nilpotent, i.e., that the entries of A satisfy equations (4.2). Certainly, $a_0 + b_0 = 1 - 1 = 0$ and

$$a_i + b_i + a_0 b_{i-1} + \cdots + a_{i-1} b_0 = a_i + a_{i-1} b_0 = 1 - 1 = 0$$

for all $i = 1, \ldots, k-1$. Also,

$$a_k + b_k + a_0 b_{k-1} + \cdots + a_{k-1} b_0 = b_k + a_{k-1} b_0 = 1 - 1 = 0.$$

Since $b_0 = -1$ and $b_j = 0$ for $j = k+1, \ldots, n-2$, on letting $b_{n-1} = 0$, the remaining equations have the form

$$0 = a_i + b_i + a_0 b_{i-1} + \cdots + a_{i-1} b_0 = a_i + a_{i-1-k} - a_{i-1},$$

where $k+1 \leq i \leq n-1$. For $k+1 \leq i \leq \min\{2k, n-1\}$,

$$a_i + a_{i-(k+1)} - a_{i-1} = k - i + 1 - (k - i + 1) = 0.$$

If $n - 1 \leq 2k$, then the proof is concluded. Suppose that $n \geq 2k+2$. The inequality

$$n < 2k + 2 + \frac{1}{2}(\sqrt{1+8k} - 1) \leq 3k + 2,$$

implies that $n-1 \leq 3k$. Thus

$$a_i + a_{i-1-k} - a_{i-1} = \frac{1}{2}(i^2 - i) + 2(k^2 - ik) + k - (i - 1 - k) - \left(\frac{1}{2}((i-1)^2 - (i-1)) + 2(k^2 - (i-1)k)\right) = 0$$

for all $i = 2k+1, \ldots, n-1$. This concludes the proof. □

To illustrate Theorem 4.3, consider the case $k = 1$. Since

$$2k + \frac{1}{2}(\sqrt{1+8k} + 3) = 5,$$

it follows from Theorem 4.3 that $\mathcal{V}_3(1)$ and $\mathcal{V}_4(1)$ exist and are minimal spectrally arbitrary patterns such that all of their superpatterns are spectrally arbitrary patterns. Since $\mathcal{V}_5^*(1)$ does not allow nilpotency, $\mathcal{V}_5(1)$ does not exist. On the other hand, $5 < 4 + \frac{1}{2}(\sqrt{17}+3)$, so $\mathcal{V}_5(2)$ exists and is equal to $\mathcal{V}_{5,2}$. Thus for $n = 5$, $\mathcal{V}_5(k)$ exists if and only if the inequality in Theorem 4.3 holds. In general, this is not true. For instance, the pattern $\mathcal{V}_8^*(2)$ allows nilpotency, as demonstrated by

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \in \mathcal{V}_8^*(2).$$

However, $n \not< 2k + \frac{1}{2}(\sqrt{1+8k}+3)$ for $n = 8$ and $k = 2$. Note also that the above sign pattern is not equal to $\mathcal{V}_{8,2}$.

A second class of sign patterns arising from patterns $\mathcal{V}_n^*(I)$ is as follows. Let $n \geq 4$, and let $\mathcal{V}_n^{\text{alt}}$ be the $n \times n$ sign pattern with positive signs in the positions

$$\left\{(4j+1, 1) : \ 0 \leq j \leq \left\lfloor \frac{n-1}{4} \right\rfloor \right\} \ \text{ and } \ \left\{(n, n-(4j+1)) : \ 0 \leq j \leq \left\lfloor \frac{n-2}{4} \right\rfloor \right\};$$

negative signs in the positions

$$\{(j, j+1) : \ 1 \leq j \leq n-1\},$$
$$\left\{(4j+3, 1) : \ 0 \leq j \leq \left\lfloor \frac{n-3}{4} \right\rfloor \right\},$$
$$\left\{(n, n-(4j+3)) : \ 0 \leq j \leq \left\lfloor \frac{n-4}{4} \right\rfloor \right\}, \ \text{ and } \ (n, n);$$

and zeros elsewhere. To illustrate,

$$\mathcal{V}_7^{\text{alt}} = \begin{bmatrix} + & - & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & - & 0 & 0 & 0 & 0 \\ - & 0 & 0 & - & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & - & 0 & 0 \\ + & 0 & 0 & 0 & 0 & - & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & - \\ - & + & 0 & - & 0 & + & - \end{bmatrix}, \quad \mathcal{V}_8^{\text{alt}} = \begin{bmatrix} + & - & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & - & 0 & 0 & 0 & 0 & 0 \\ - & 0 & 0 & - & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & - & 0 & 0 & 0 \\ + & 0 & 0 & 0 & 0 & - & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & - & 0 \\ - & 0 & 0 & 0 & 0 & 0 & 0 & - \\ - & 0 & + & 0 & - & 0 & + & - \end{bmatrix}.$$

THEOREM 4.4. *For $n \geq 4$, let I consist of all odd integers $i \leq n-2$. Then $\mathcal{V}_n(I)$ exists and is identical to $\mathcal{V}_n^{alt}$. Furthermore, it is a minimal spectrally arbitrary pattern, and any superpattern of $\mathcal{V}_n(I)$ is spectrally arbitrary.*

Proof. Let A be as in (4.1), and assume that n is odd. Let $a_i = b_j = 0$ for all odd $i \leq n-2$ and all even j such that $2 \leq j \leq n-2$, let $b_0 = -1$, and define $b_{n-1} = 0$. For all $0 \leq i \leq \frac{n-1}{2}$, let $a_{2i} = (-1)^i C_i$, where $C_i = \frac{1}{i+1}\binom{2i}{i}$ is the ith Catalan number (see, for example, [10, 11] and note that $C_0 = 1$). Also, let $b_{2i+1} = a_{2i} = (-1)^i C_i$ for all $0 \leq i \leq \frac{n-3}{2}$. Then $A \in \mathcal{V}_n^*(I)$ and $A \in \mathcal{V}_n^{\text{alt}}$. To conclude the proof, it is sufficient, by Lemma 4.1, to show that A is nilpotent. Certainly, $a_0 + b_0 = 0$. For each $i \geq 0$, the Catalan number C_{i+1} satisfies the recursive identity

$$C_{i+1} = \sum_{j=0}^{i} C_j C_{i-j}$$

(see [11, p. 117]). Thus, for $0 \leq i \leq \frac{n-3}{2}$,

$$a_{2i+1} + b_{2i+1} + a_{2i}b_0 + \cdots + a_0 b_{2i} = b_{2i+1} - a_{2i} = 0$$

and

$$a_{2i+2} + b_{2i+2} + a_{2i+1}b_0 + \cdots + a_0 b_{2i+1} = a_{2i+2} + \sum_{j=0}^{i} a_{2j}a_{2(i-j)} =$$

$$(-1)^{i+1}C_{i+1} + \sum_{j=0}^{i}(-1)^j C_j (-1)^{i-j} C_{i-j} = (-1)^i\left(- C_{i+1} + \sum_{j=0}^{i} C_j C_{i-j}\right) = 0.$$

The equations (4.2) are all satisfied, so A is nilpotent.

Assume that n is even. Let $a_i = b_j = 0$ for all odd $i \leq n-2$ and all even j such that $2 \leq j \leq n-2$, and let $b_0 = -1$. For all $0 \leq i \leq \frac{n-2}{2}$, let $a_{2i} = (-1)^i C_i$. Let $a_{n-1} = a_{n-2}$, and let $b_{2i+1} = a_{2i} = (-1)^i C_i$ for all $0 \leq i \leq \frac{n-4}{2}$. Then $A \in \mathcal{V}_n^*(I)$ and $A \in \mathcal{V}_n^{\text{alt}}$. To conclude the proof, it is sufficient, by Lemma 4.1, to show that A is nilpotent. Certainly, $a_0 + b_0 = 0$. For $0 \leq i \leq \frac{n-4}{2}$,

$$a_{2i+1} + b_{2i+1} + a_{2i}b_0 + \cdots + a_0 b_{2i} = b_{2i+1} - a_{2i} = 0$$

and

$$a_{2i+2} + b_{2i+2} + a_{2i+1}b_0 + \cdots + a_0 b_{2i+1} = a_{2i+2} + \sum_{j=0}^{i} a_{2j}a_{2(i-j)}$$

$$= (-1)^{i+1}C_{i+1} + \sum_{j=0}^{i}(-1)^j C_j (-1)^{i-j} C_{i-j} = (-1)^i\left(- C_{i+1} + \sum_{j=0}^{i} C_j C_{i-j}\right) = 0.$$

Furthermore,

$$a_{n-1} + a_{n-2}b_0 + \cdots + a_0 b_{n-2} = a_{n-1} - a_{n-2} = 0.$$

The equations (4.2) are all satisfied, so A is nilpotent. ∎

5. All minimal 3×3 spectrally arbitrary patterns. In the proof of Theorem 5.2, it will be shown that a 3×3 irreducible sign pattern (with at least one positive and one negative diagonal entry) is spectrally arbitrary if and only if it allows nilpotency. Our approach to deciding whether or not a 3×3 sign pattern allows nilpotency is different and more explicit than that in [3, Theorem 4.1]. First, the following lemma is given, which precludes certain 3×3 patterns from allowing nilpotency.

LEMMA 5.1. *Let $\mathcal{A}$ be the sign pattern determined by any $n \times n$ matrix A with nonzero entries a_{ii} for $i = 1, \ldots, n$; $a_{i,i+1}$ for $i = 1, \ldots, n-1$; and a_{n1} (i.e., $D(\mathcal{A})$ is a directed n-cycle with a loop at each vertex). Then $\mathcal{A}$ allows nilpotency if and only if $n = 2$.*

Proof. The characteristic equation of A is

$$0 = \lambda^n - \sum_{i=1}^{n} a_{ii}\lambda^{n-1} + \sum_{1\le i<j\le n} a_{ii}a_{jj}\lambda^{n-2} - \cdots + (-1)^n \prod_{i=1}^{n} a_{ii} - a_{n1}\prod_{i=1}^{n-1} a_{i,i+1}.$$

If A is to be nilpotent, then

$$\begin{aligned} 0 &= \sum_{i=1}^{n} a_{ii}, \\ 0 &= \sum_{1\le i<j\le n} a_{ii}a_{jj}, \\ &\vdots \\ 0 &= \sum_{1\le i_1<i_2<\cdots<i_{n-1}\le n} a_{i_1i_1}a_{i_2i_2}\cdots a_{i_{n-1}i_{n-1}}. \end{aligned}$$

The a_{ii} are roots of the equation $(x - a_{11})(x - a_{22})\cdots(x - a_{nn}) = 0$, which is

$$x^n + (-1)^n \prod_{i=1}^{n} a_{ii} = 0$$

by the above equations.

If $n = 2$, then this can be satisfied with the two real numbers $\pm\sqrt{|a_{11}a_{22}|}$, i.e., $a_{11} = a$ and $a_{22} = -a$. But for $n \ge 3$, the equation cannot be satisfied for n real values, thus $\mathcal{A}$ does not allow nilpotency. ∎

THEOREM 5.2. *The family of 3×3 minimal spectrally arbitrary sign patterns consists of the sign patterns that are equivalent to one of the patterns $\mathcal{T}_3$, $\mathcal{U}_3$, $\mathcal{V}_3$, and $\mathcal{W}_3$ in Figure 5.1. Furthermore, every 3×3 spectrally arbitrary sign pattern is equivalent to a superpattern of one of these four patterns.*

$$\mathcal{T}_3 = \begin{bmatrix} + & - & 0 \\ + & 0 & - \\ 0 & + & - \end{bmatrix} \qquad \mathcal{U}_3 = \begin{bmatrix} + & - & + \\ + & - & 0 \\ + & 0 & - \end{bmatrix} \qquad \mathcal{V}_3 = \begin{bmatrix} + & - & 0 \\ + & 0 & - \\ + & 0 & - \end{bmatrix} \qquad \mathcal{W}_3 = \mathcal{W}_3(1) = \begin{bmatrix} + & + & - \\ + & 0 & - \\ + & 0 & - \end{bmatrix}$$

FIG. 5.1. *The minimal 3×3 spectrally arbitrary patterns.*

Proof. In [1], it is shown that $\mathcal{T}_3$ and $\mathcal{U}_3$ are minimal spectrally arbitrary patterns and that each superpattern of these two patterns is also spectrally arbitrary. By Theorem 2.5, Proposition 3.1, and the comments following Corollary 3.2, the patterns $\mathcal{V}_3$ and $\mathcal{W}_3$ are both minimal spectrally arbitrary patterns, and each superpattern of these two patterns is also spectrally arbitrary. Since it is easily shown that there are no reducible 3×3 spectrally arbitrary patterns, to conclude the proof it is necessary to demonstrate that the patterns that are equivalent to these four patterns and their superpatterns are the only (irreducible) 3×3 spectrally arbitrary patterns. This is done by proving that each 3×3 sign pattern not equivalent to any superpattern of $\mathcal{T}_3$, $\mathcal{U}_3$, $\mathcal{V}_3$, or $\mathcal{W}_3$ does not allow nilpotency and, thus, is not a spectrally arbitrary pattern. There are many such sign patterns and a detailed account for each pattern would be quite tedious. Fortunately, this number can be reduced as follows. Up to equivalence, each irreducible 3×3 spectrally arbitrary pattern has one of the following forms:

$$\begin{bmatrix} + & \# & \# \\ + & \# & \# \\ 0 & + & - \end{bmatrix} \qquad \begin{bmatrix} + & \# & \# \\ + & \# & \# \\ + & \# & - \end{bmatrix},$$

where each # denotes either a plus, minus, or zero entry. Of $3^4 + 3^5 = 324$ possible sign patterns, 78 are reducible and 115 are equivalent to superpatterns of one or more of the patterns $\mathcal{T}_3$, $\mathcal{U}_3$, $\mathcal{V}_3$, and $\mathcal{W}_3$. Of the remaining 131 patterns, there are 71 patterns $\mathcal{A}$ such that for any matrix $A \in \mathcal{A}$, the characteristic polynomial $p_A(x) = x^3 - \alpha_1 x^2 + \alpha_2 x - \alpha_3$ contains a coefficient α_1, α_2, or α_3 that has a fixed sign, regardless of the specific matrix A. Such patterns cannot allow nilpotency.

The remaining 60 patterns fall into four general classes described below. By Lemma 2.3, it may be assumed that any two of the nonzero strictly upper triangular entries of any given irreducible 3×3 matrix both have magnitude 1.

The first of the four classes consists of the four patterns

$$\begin{bmatrix} + & 0 & - \\ + & - & 0 \\ 0 & + & - \end{bmatrix}, \begin{bmatrix} + & 0 & + \\ + & - & 0 \\ 0 & + & - \end{bmatrix}, \begin{bmatrix} + & 0 & - \\ + & + & 0 \\ 0 & + & - \end{bmatrix}, \text{ and } \begin{bmatrix} + & 0 & + \\ + & + & 0 \\ 0 & + & - \end{bmatrix}.$$

By Lemma 5.1, such sign patterns do not allow nilpotency.

For the second class, consider

$$\mathcal{A} = \begin{bmatrix} + & + & + \\ + & + & + \\ + & + & - \end{bmatrix} \quad \text{with} \quad A = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & -j \end{bmatrix} \in \mathcal{A}.$$

The matrix A has the characteristic polynomial

$$\begin{aligned} p_A(x) = x^3 &+ (j - a - e)x^2 + (ae - aj - bd - cg - ej - fh)x \\ &+ aej + ahf - bdj - bgf - cdh + cge. \end{aligned}$$

Assuming that $\mathcal{A}$ allows nilpotency, then values of $a, b, \ldots, j$ exist such that A is nilpotent, i.e., $p_A(x) = x^3$. In this case, $j = a + e$, which implies that

$$\begin{aligned} 0 &= ae - aj - bd - cg - ej - fh \\ &= -a^2 - ae - bd - cg - e^2 - fh < 0, \end{aligned}$$

a contradiction. Thus, $\mathcal{A}$ does not allow nilpotency. The same conclusion is valid if one or more of the entries b, c, d, f, g, or h is equal to 0, and/or both b and d, both c and g, and/or both f and h are nonpositive. These sign patterns and their equivalent patterns account for 35 of the remaining 60 patterns.

For the third sign pattern class, consider

$$\mathcal{A} = \begin{bmatrix} + & - & - \\ + & - & + \\ + & + & - \end{bmatrix} \quad \text{with} \quad A = \begin{bmatrix} a & -d & -1 \\ b & -e & h \\ c & f & -j \end{bmatrix} \in \mathcal{A}.$$

The matrix A has the characteristic polynomial

$$\begin{aligned} p_A(x) = x^3 &+ (e + j - a)x^2 + (bd + cd + ej - ae - aj - fh)x \\ &+ afh + bf + bdj + ce + ch - aej. \end{aligned}$$

To show that $\mathcal{A}$ does not allow nilpotency, assume that $p_A(x) = x^3$ for appropriate values of $a, b, \ldots, j$. It must hold that $a = e + j$, so

$$c = ae + aj + fh - bd - ej = e^2 + ej + fh + j^2 - bd.$$

Thus the constant term gives

$$0 = -bd^2h - bde + e^3 + bdj + bf + de^2h + dehj + dfh^2 + dhj^2 + 2efh + fhj,$$

so

$$b = \frac{de^2h + dehj + dfh^2 + dhj^2 + e^3 + 2efh + fhj}{d^2h + de - dj - f}.$$

Since $a, b, \ldots, j > 0$, it follows that $d^2h + de - dj - f > 0$. However,

$$\begin{aligned} c &= e^2 + ej + fh + j^2 - bd \\ &= \frac{-defh - 2dfhj - dj^3 - e^2f - efj - f^2h - fj^2}{d^2h + de - dj - f} < 0, \end{aligned}$$

a contradiction, so $\mathcal{A}$ does not allow nilpotency. The same arguments are valid if any of d, f, and h equal 0 such that $d + f > 0$. These sign patterns and their equivalent patterns account for 8 of the 60 patterns.

For the fourth class, let

$$\mathcal{A} = \begin{bmatrix} + & - & 0 \\ + & - & + \\ + & + & - \end{bmatrix} \quad \text{with} \quad A = \begin{bmatrix} a & -1 & 0 \\ b & -e & 1 \\ c & f & -j \end{bmatrix} \in \mathcal{A}.$$

Assuming that $\mathcal{A}$ allows nilpotency, it is possible to assign values to $a, b, \ldots, j$ such that the characteristic polynomial

$$p_A(x) = x^3 + (e + j - a)x^2 + (b + ej - ae - aj - f)x + af + bj + c - aej$$

equals x^3. If this is true, then $a = j + e$, so

$$b = ae + aj - ej + f = e^2 + j^2 + ej + f$$

and

$$0 = af + bj + c - aej = c + ef + 2fj + j^3 > 0,$$

a contradiction. Thus, $\mathcal{A}$ does not allow nilpotency. The same arguments and conclusion are true if c or f equals 0. The cases

$$\begin{bmatrix} + & - & - \\ + & - & 0 \\ 0 & + & - \end{bmatrix}, \begin{bmatrix} + & - & - \\ + & - & + \\ 0 & + & - \end{bmatrix}, \begin{bmatrix} + & + & 0 \\ + & + & - \\ 0 & + & - \end{bmatrix}, \text{ and } \begin{bmatrix} + & + & + \\ + & + & - \\ 0 & + & - \end{bmatrix}$$

are proven to not allow nilpotency in the same way. The sign patterns above and their equivalent patterns account for 13 of the 60 patterns.

It may be verified by inspection that every one of the 60 sign pattern belongs to one of the four classes above, no members of which allow nilpotency. This concludes the proof. □

6. Concluding remarks. Since our interest is on minimal spectrally arbitrary patterns, we address the question of the least number of nonzero entries required by such a pattern.

CONJECTURE 6.1. *For $n \geq 2$, an $n \times n$ sign pattern that is spectrally arbitrary has at least $2n$ nonzero entries.*

Conjecture 6.1 is verified for $n = 2$ by Example 2.2, and Theorem 5.2 verifies the conjecture for $n = 3$ (since there are no 3×3 reducible spectrally arbitrary sign patterns). For all $n \geq 3$, this bound is realized by $\mathcal{V}_n$ (Theorem 2.4). It is also realized by the antipodal tridiagonal sign pattern T_n in [1, 2] for all values of n for which T_n is known to be spectrally arbitrary (i.e., $2 \leq n \leq 16$).

Let $\mathbb{Q}[X]$ be the set of polynomials with rational coefficients and finite degree. A set $S \subseteq \mathbb{R}$ is algebraically independent if, for all $s_1, \ldots, s_n \in S$ and each nonzero polynomial $p(x_1, \ldots, x_n) \in \mathbb{Q}[X]$, $p(s_1, \ldots, s_n) \neq 0$ (see [6, p. 316] for further details). Let $\mathbb{Q}(S)$ denote the field of rational expressions

$$\left\{ \frac{p(s_1, \ldots, s_m)}{q(t_1, \ldots, t_n)} \ : \ p(x_1, \ldots, x_m), q(x_1, \ldots, x_n) \in \mathbb{Q}[X], s_1, \ldots, s_m, t_1, \ldots, t_n \in S \right\},$$

and let the *transcendental degree* of S be

$$tr.d.S = \sup\{|T| \ : \ T \subseteq S, \ T \text{ is algebraically independent}\}.$$

The following theorem very nearly verifies Conjecture 6.1.

THEOREM 6.2. *For $n \geq 2$, an irreducible $n \times n$ sign pattern that is spectrally arbitrary has at least $2n - 1$ nonzero entries.*

Proof. Let $\mathcal{A}$ be an irreducible $n \times n$ spectrally arbitrary sign pattern with $n_{\mathcal{A}}$ nonzero entries. Choose a set $\{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{R}$ that is algebraically independent. By Lemma 2.3, $\mathcal{A}$ has a realization $A = [a_{ij}]$ with characteristic polynomial

$$p_A(x) = x^n - \alpha_1 x^{n-1} + \cdots + (-1)^n \alpha_n$$

and $n - 1$ (off-diagonal) entries with magnitude 1. Since for each $1 \leq i \leq n$, α_i is a polynomial in the entries $\{a_{ij} : 1 \leq i, j \leq n\}$ with rational coefficients, it follows that $\mathbb{Q}(\alpha_1, \ldots, \alpha_n) \subseteq \mathbb{Q}(a_{ij} : 1 \leq i, j \leq n)$, so

$$n = tr.d.\mathbb{Q}(\alpha_1, \ldots, \alpha_n) \leq tr.d.\mathbb{Q}(a_{ij} : 1 \leq i, j \leq n) \leq n_{\mathcal{A}} - (n - 1).$$

Thus, $n_{\mathcal{A}} \geq 2n-1$. □

It is clear from the proof of Theorem 5.2 that a 3×3 irreducible sign pattern (with at least one positive and one negative diagonal entry) allows nilpotency if and only if it is a spectrally arbitrary pattern. This is not generally true, as the following 4×4 sign pattern demonstrates. Let

$$\mathcal{A} = \begin{bmatrix} + & + & 0 & 0 \\ 0 & 0 & + & 0 \\ 0 & - & 0 & + \\ - & 0 & 0 & - \end{bmatrix} \quad \text{with} \quad A = \begin{bmatrix} a & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -c & 0 & 1 \\ -b & 0 & 0 & -d \end{bmatrix} \in \mathcal{A}.$$

By Lemma 2.3, it may be assumed without loss of generality that each realization of $\mathcal{A}$ has the form of A above. The characteristic polynomial of A is

$$p_A(x) = x^4 - (a-d)x^3 - (ad-c)x^2 - (a-d)cx - acd + b.$$

If $(a-d)c = 0$, then $a - d = 0$, so $p_A(x)$ cannot equal $x^4 - \alpha x^3$ for any nonzero α. Thus, $\mathcal{A}$ does not allow the spectrum $\{0, 0, 0, \alpha\}$ for any nonzero α, and thus $\mathcal{A}$ is not spectrally arbitrary. However, $\mathcal{A}$ does allow nilpotency, since A is nilpotent for $a = b = c = d = 1$.

REFERENCES

[1] J. H. Drew, C. R. Johnson, D. D. Olesky, and P. van den Driessche, *Spectrally arbitrary patterns*, Linear Algebra Appl., 308 (2000), pp. 121–137.

[2] L. Elsner, D. D. Olesky, and P. van den Driessche, *Low rank perturbations and the spectrum of a tridiagonal sign pattern*, Linear Algebra Appl., 374 (2003), pp. 219–230.

[3] C. Eschenbach and Z. Li, *Potentially nilpotent sign pattern matrices*, Linear Algebra Appl., 299 (1999), pp. 81–99.

[4] Y. Gao and J. Li, *On the potential stability of star sign pattern matrices*, Linear Algebra Appl., 327 (2001), pp. 61–68.

[5] Y. Gao and Y. Shao, *Inertially arbitrary patterns*, Linear Multilinear Algebra, 49 (2001), pp. 161–168.

[6] T. Hungerford, *Algebra*, 2nd ed., Graduate Texts in Math., 73, Springer-Verlag, New York-Berlin, 1980.

[7] C. R. Johnson and T. A. Summers, *The potentially stable tree sign patterns for dimensions less than five*, Linear Algebra Appl., 126 (1989), pp. 1–13.

[8] J. J. McDonald, D. D. Olesky, M. J. Tsatsomeros, and P. van den Driessche, *On the spectra of striped sign patterns*, Linear Multilinear Algebra, 51 (2003), pp. 39–48.

[9] Z. Miao and J. Li, *Inertially arbitrary* $(2r-1)$*-diagonal sign patterns*, Linear Algebra Appl., 357 (2002), pp. 133–141.

[10] R. P. Stanley, *Enumerative Combinatorics,* Vol. 2, Cambridge University Press, Cambridge, UK, 1999.

[11] J. H. van Lint and R. M. Wilson, *A Course in Combinatorics*, Cambridge University Press, Cambridge, UK, 1992.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 1, pp. 272–284

A DUAL APPROACH TO SEMIDEFINITE LEAST-SQUARES PROBLEMS*

JÉRÔME MALICK†

Abstract. In this paper, we study the projection onto the intersection of an affine subspace and a convex set and provide a particular treatment for the cone of positive semidefinite matrices. Among applications of this problem is the calibration of covariance matrices. We propose a Lagrangian dualization of this least-squares problem, which leads us to a convex differentiable dual problem. We propose to solve the latter problem with a quasi-Newton algorithm. We assess this approach with numerical experiments which show that fairly large problems can be solved efficiently.

Key words. Lagrangian duality, semidefinite optimization, calibration of covariance matrices

AMS subject classifications. 65K05, 65F99, 90C22, 91B28

DOI. 10.1137/S0895479802413856

1. Introduction.

1.1. To find the best approximation. We propose a method to solve the following problem: to project a point, in a Euclidean space, onto the intersection of a closed convex set $\mathcal{K}$ and of an affine subspace. We are particularly interested in the case where $\mathcal{K}$ is a cone, more specifically the cone of symmetric positive semidefinite matrices. We call this latter problem *semidefinite least-squares (sdls)*.

Semidefinite least-squares problems arise in different fields of numerical and applied mathematics. For instance, a "good" approximation of a covariance matrix between n assets, which plays a key role in portfolio risk analysis, could be obtained from a first estimate by solving a semidefinite least-squares (this is developed in subsection 5.4). Semidefinite least-squares also occur in robust quadratic optimization and numerical linear algebra (preconditioning of linear system and error analysis of Jacobi methods for the symmetric eigenvalue problem; see [DH00]).

Our aim is to propose an algorithm based on Lagrangian duality to solve the above-mentioned least-squares problem. This paper is organized as follows. We focus, in section 2, on the case where there are no affine constraints: using tools from convex analysis, we recover known properties of distance functions. In section 3, we introduce affine constraints and we show that their dualization yields a dual problem which is convex and differentiable. A quasi-Newton algorithm is proposed in section 4 to solve this last problem. Computational results, comparison with existing methods, and applications of the semidefinite version of this algorithm are presented in section 5.

1.2. Basic notation. The general framework of this paper is a Euclidean space, say $\mathbb{R}^p$, equipped with a scalar product $\langle \cdot, \cdot \rangle$. We will denote by $|| \cdot ||$ the associated norm. We consider, in particular, the space of $n \times n$ symmetric matrices $\mathcal{S}_n$, equipped, for instance, with the Frobenius scalar product

$$\forall X, Y \in \mathcal{S}_n \quad \langle X, Y \rangle \,=\, \mathrm{tr}(XY) \,=\, \sum_{i,j=1}^{n} X_{ij} Y_{ij}\,,$$

*Received by the editors September 3, 2002; accepted for publication (in revised form) by M. L. Overton November 14, 2003; published electronically September 14, 2004.
http://www.siam.org/journals/simax/26-1/41385.html
†INRIA, 655 av. de l'Europe, Montbonnot, 38334 Saint Ismier, France (jerome.malick@inria.fr).

where $\mathrm{tr}(X)$ is the trace of the matrix X. We give below a short glossary of symbols:

- the closed convex cone of positive semidefinite matrices is denoted by $\mathcal{S}_n^+$; we use the notation $X \succeq 0$ to express that X lies in $\mathcal{S}_n^+$;
- the adjoint of a linear mapping $\mathcal{A}$ is denoted by $\mathcal{A}^*$;
- for any vector x in $\mathbb{R}^n$, $\mathrm{Diag}\, x$ denotes the diagonal matrix with the vector x on the main diagonal; its adjoint operator diag: $\mathcal{S}_n \to \mathbb{R}^n$ is $\mathrm{diag}(A) = [a_{11}, \ldots, a_{nn}]^\top$.

1.3. Formulation. The problems we will focus on can be expressed as follows. Let $\mathcal{K}$ be a closed convex set of $\mathbb{R}^p$. Let a vector $b \in \mathbb{R}^m$ and a linear operator $\mathcal{A} : \mathbb{R}^p \longrightarrow \mathbb{R}^m$ be given. We want to compute the projection of a vector $c \in \mathbb{R}^p$ onto the closed convex subset of $\mathbb{R}^p$ formed by the intersection of $\mathcal{K}$ and the affine subspace defined by $\mathcal{A}$ and b. Our goal is to design an algorithm to solve

$$\begin{cases} \inf \quad \frac{1}{2}||x - c||^2, \\ \mathcal{A}\, x = b\,, \\ x \in \mathcal{K}\,. \end{cases} \tag{1.1}$$

Each component function of $\mathcal{A}$ can be expressed as a scalar product: there exist m elements $a_i \in \mathbb{R}^p$ such that $\mathcal{A}(x) = [\langle a_1, x\rangle, \ldots, \langle a_m, x\rangle]^\top$. Therefore an equivalent formulation is (for b, a_i, and c given)

$$\begin{cases} \inf \quad \frac{1}{2}||x - c||^2, \\ \langle a_i, x\rangle = b_i\,, \quad i = 1, \ldots, m, \\ x \in \mathcal{K}\,. \end{cases}$$

The first remark is that, if the feasible domain is nonempty, there exists a unique $x^\star$ which achieves the above infimum. In what follows, we assume this to be the case; therefore we use the notation min rather than inf for this least-squares problem.

To end this introduction, we specify the framework of this paper. Our first motivation is to solve efficiently semidefinite least-squares (i.e., when $\mathcal{K} = \mathcal{S}_n^+$), which section 5 is devoted to. Although the material of this paper can be developed with a general closed convex set $\mathcal{K}$ (see Remarks 2.3 and 4.3(ii)), we restrict ourselves to the case where $\mathcal{K}$ is a *closed convex cone*. This allows us to introduce adapted tools, to simplify calculus and to stay closer to semidefinite least-squares.

2. Projection onto a closed convex cone. To begin with, we isolate the problem of computing the projection $p_{\mathcal{K}}(c)$ of a fixed $c \in \mathbb{R}^p$ onto a closed convex cone $\mathcal{K}$, with a special study for $\mathcal{K} = \mathcal{S}_n^+$. The aim of this section is twofold:

(1) to recall results we will need;

(2) to draw connections between these results and tools from convex analysis.

2.1. Moreau theorem and Moreau regularization. The projection onto a cone $\mathcal{K}$ enjoys properties which come close to those of the projection onto a subspace. The set playing the role of the orthogonal subspace is the *polar cone* $\mathcal{K}^o$ of $\mathcal{K}$:

$$\mathcal{K}^o := \{s \in \mathbb{R}^p : \quad \langle s, x\rangle \leq 0 \quad \text{for all } x \in \mathcal{K}\}\,.$$

A first observation is that $\mathcal{K}^o$ is also closed and convex. There is a decomposition result which generalizes the decomposition of a vector space as the direct sum of a (closed) subspace and its orthogonal (see [HUL01, Chap. A]).

THEOREM 2.1 (Moreau decomposition). *Let $\mathcal{K}$ be a closed convex cone. For the three elements x, x_1, and x_2 in $\mathbb{R}^p$, the two properties below are equivalent:*

(i) $x = x_1 + x_2$ *with* $x_1 \in \mathcal{K}$, $x_2 \in \mathcal{K}^o$ *and* $\langle x_1, x_2 \rangle = 0$,
(ii) $x_1 = p_{\mathcal{K}}(x)$ *and* $x_2 = p_{\mathcal{K}^o}(x)$.

We turn now to variational properties of the half-squared distance to $\mathcal{K}$, which will be needed in section 3:

$$\begin{array}{rccl} d_{\mathcal{K}} & : \mathbb{R}^p & \longrightarrow & \mathbb{R}, \\ & x & \longmapsto & \min\limits_{y \in \mathcal{K}} \ \frac{1}{2}||x-y||^2 . \end{array} \tag{2.1}$$

We start by observing that there is another useful expression of $d_{\mathcal{K}}$. By definition, the above minimum is reached at the unique point $p_{\mathcal{K}}(x)$. Then Theorem 2.1 yields

$$d_{\mathcal{K}}(x) \ = \ \frac{1}{2}||x - p_{\mathcal{K}}(x)||^2 \ = \ \frac{1}{2}||p_{\mathcal{K}^o}(x)||^2 . \tag{2.2}$$

The following properties are not new, and can be proved with basic tools. Here, we show that they are straightforward applications of properties of *Moreau–Yosida regularization* [HUL93, Chap. XV]. For a convex function f on $\mathbb{R}^p$, we define the Moreau–Yosida regularization of f to be the function

$$x \longmapsto \min_{y \in \mathbb{R}^p} \ \left\{ f(y) + \frac{1}{2}||x-y||^2 \right\} .$$

THEOREM 2.2. *Let $\mathcal{K}$ be a closed convex cone in $\mathbb{R}^p$. Then the function $d_{\mathcal{K}}(x)$ defined by* (2.1) *is a convex differentiable function from $\mathbb{R}^p$ to $\mathbb{R}$, whose gradient is*

$$\nabla d_{\mathcal{K}}(x) = p_{\mathcal{K}^o}(x) . \tag{2.3}$$

Furthermore the gradient function is 1*-Lipschitz continuous.*

Proof. Let $\mathrm{I}_{\mathcal{K}}$ be the indicator function of $\mathcal{K}$ (whose values are 0 on $\mathcal{K}$ and $+\infty$ elsewhere). The theorem is just Theorem 4.1.4 of [HUL93, Chap. XV] written in our case, since $d_{\mathcal{K}}$ can be interpreted as the Moreau–Yosida regularization of $\mathrm{I}_{\mathcal{K}}$:

$$d_{\mathcal{K}}(x) = \min_{y \in \mathbb{R}^p} \ \left\{ \mathrm{I}_{\mathcal{K}}(y) + \frac{1}{2}||x-y||^2 \right\} .$$

We get, in particular, $\nabla d_{\mathcal{K}}(x) = x - p_{\mathcal{K}}(x) = p_{\mathcal{K}^o}(x)$ (by Theorem 2.1). The Lipschitz property is clear here since the gradient is a projection. □

Remark 2.3. Note that the above result is valid when k is a general closed convex set, but then the expression (2.3) of the gradient is replaced by $\nabla d_{\mathcal{K}}(x) = x - p_{\mathcal{K}}(x)$ which is again 1-Lipschitz.

2.2. Projection onto $\mathcal{S}_n^+$. In this subsection, we consider the semidefinite least-squares problem without any affine constraint. We recall a crucial theorem for our purposes: an explicit formula for the projection onto $\mathcal{S}_n^+$.

We need more notation. We denote by $\lambda_i(C)$ the (real) eigenvalues of $C \in \mathcal{S}_n$, and rank them in nonincreasing order

$$\lambda_1(C) \geq \lambda_2(C) \geq \cdots \geq \lambda_n(C);$$

$\lambda(C)$ will stand for $[\lambda_1(C), \ldots, \lambda_n(C)]^\top$. The symmetric matrix C is diagonalizable in an orthonormal basis of $\mathbb{R}^n$ formed by eigenvectors of C: $C = P_C(\mathrm{Diag}\lambda(C))P_C^\top$.

We will often drop the dependence on C from our notation. We denote by C_+ the "positive semidefinite part" of C (negative eigenvalues are set to zero):

$$C_+ := P \begin{bmatrix} \max\{\lambda_1, 0\} & & \\ & \ddots & \\ & & \max\{\lambda_n, 0\} \end{bmatrix} P^\top ; \tag{2.4}$$

likewise the "negative semidefinite part" is denoted by C_-.

THEOREM 2.4. *Let $C \in \mathcal{S}_n$. Then the projection $p_{\mathcal{S}_n^+}(C)$ of C onto $\mathcal{S}_n^+$ is the matrix C_+, defined by* (2.4). *Likewise the projection $p_{\mathcal{S}_n^-}(C)$ of C onto the polar cone $(\mathcal{S}_n^+)^o = \mathcal{S}_n^-$ is C_-.*

A direct proof of this result is proposed in [Hig88]. It is worth mentioning that this theorem is also a straightforward application of Theorem 2.1 (see [HUL01, Exercise A.15]), the key being that $C_+ \in \mathcal{S}_n^+$, $C_- \in \mathcal{S}_n^-$ and $\langle C_+, C_- \rangle = 0$.

Remark 2.5. The space $\mathcal{S}_n$ is frequently equipped with a weighted version of the Frobenius norm

$$||X||_W = ||W^{1/2} X W^{1/2}||,$$

where W is a positive definite matrix. It is easy to express the projection (in the sense of the weighted scalar product) of $C \in \mathcal{S}_n$ onto $\mathcal{S}_n^+$ as

$$W^{-1/2}(W^{1/2} C W^{1/2})_+ W^{-1/2}.$$

3. Lagrangian duality. We propose in this section a Lagrangian dualization of (1.1). The idea is to treat in two different ways the two different kinds of constraints: on one hand affine constraints in $\mathbb{R}^p$ and on the other hand convex constraints. The technique is to dualize only affine constraints, forming a partial Lagrangian.

All the present paper relies upon the next statement. It motivates the developments of previous sections and will give birth to computational methods.

THEOREM 3.1. *Consider the following least-squares problem in $(\mathbb{R}^p, ||\cdot||)$:*

$$(primal) \begin{cases} \min & \frac{1}{2}||x - c||^2, \\ x \in \mathcal{K}, & \mathcal{A}\, x = b\,, \end{cases}$$

which is our primal problem. Form the partial Lagrangian depending on two variables (the primal variable x which lies in $\mathcal{K} \subset \mathbb{R}^p$ and the dual variable y which lies in the constraint space $\mathbb{R}^m$)

$$L(x; y) := \frac{1}{2}||c - x||^2 - y^\top (\mathcal{A}x - b)\,. \tag{3.1}$$

Define the corresponding dual function

$$\theta(y) := \min_{x \in \mathcal{K}} L(x; y) \tag{3.2}$$

and the dual problem on the constraint space $\mathbb{R}^m$

$$(dual) \begin{cases} \sup & \theta(y), \\ y \in \mathbb{R}^m\,. \end{cases}$$

The dual function has the following expressions:

$$\begin{aligned} \theta(y) &= -\tfrac{1}{2}||p_{\mathcal{K}}(c + \mathcal{A}^* y)||^2 + \tfrac{1}{2}||c||^2 + y^\top b \\ &= -d_{\mathcal{K}^o}(c + \mathcal{A}^* y) + \tfrac{1}{2}||c||^2 + y^\top b\,. \end{aligned} \tag{3.3}$$

Proof. Let us transform the partial Lagrangian to isolate the function $d_{\mathcal{K}}$ of (2.1):

$$\begin{aligned} L(x;y) &= \tfrac{1}{2}||c-x||^2 - \langle \mathcal{A}^* y, x\rangle + y^\top b \\ &= \tfrac{1}{2}||(c+\mathcal{A}^* y) - x||^2 - (\tfrac{1}{2}||\mathcal{A}^* y||^2 + \langle c, \mathcal{A}^* y\rangle) + y^\top b \\ &= \tfrac{1}{2}||(c+\mathcal{A}^* y) - x||^2 - (\tfrac{1}{2}||\mathcal{A}^* y + c||^2 - \tfrac{1}{2}||c||^2) + y^\top b\,. \end{aligned}$$

Now get an expression of θ. For any fixed $y \in \mathbb{R}^m$:

$$\begin{aligned} \theta(y) &:= \min_{x\in\mathcal{K}} L(x;y) \\ &= d_{\mathcal{K}}(c+\mathcal{A}^* y) - \tfrac{1}{2}||\mathcal{A}^* y + c||^2 + \tfrac{1}{2}||c||^2 + y^\top b\,. \end{aligned}$$

Simplify with (2.2):

$$\begin{aligned} \theta(y) &= \tfrac{1}{2}||p_{\mathcal{K}^o}(c+\mathcal{A}^* y)||^2 - \tfrac{1}{2}||\mathcal{A}^* y + c||^2 + \tfrac{1}{2}||c||^2 + y^\top b \\ &= -\tfrac{1}{2}||p_{\mathcal{K}}(c+\mathcal{A}^* y)||^2 + \tfrac{1}{2}||c||^2 + y^\top b \\ &= -d_{\mathcal{K}^o}(c+\mathcal{A}^* y) + \tfrac{1}{2}||c||^2 + y^\top b\,. \end{aligned}$$

We therefore obtain the expected formulations of θ. ∎

Notice that we know the unique point in $\mathcal{K}$ which achieves the minimum in (3.2) for $y \in \mathbb{R}^p$. In the remainder of the paper, we denote it by $x(y)$:

$$x(y) := \operatorname*{argmin}_{x\in\mathcal{K}} L(x;y) = p_{\mathcal{K}}(c+\mathcal{A}^* y)\,. \tag{3.4}$$

In other words there holds

$$\theta(y) = L(x(y);y)\,. \tag{3.5}$$

The dual function θ inherits the properties of $d_{\mathcal{K}^o}$ studied in section 2.

THEOREM 3.2. *The function θ of* (3.3) *satisfies the properties below:*

(i) *θ is concave,*

(ii) *θ is differentiable,*

(iii) *$\nabla\theta$ is Lipschitz continuous and is given by*

$$\nabla\theta(y) = -\mathcal{A}\{p_{\mathcal{K}}(c+\mathcal{A}^* y)\} + b \tag{3.6}$$

Proof. The dual function, as a minimum of affine functions of y, is concave by construction. Besides, with equation (3.3), according to results on $d_{\mathcal{K}^o}$ (Theorem 2.2 for $\mathcal{K}^o$), θ is differentiable, its gradient is

$$\begin{aligned} \nabla\theta(y) &= -\mathcal{A}\{\nabla d_{\mathcal{K}^o}(c+\mathcal{A}^* y)\} + b \\ &= -\mathcal{A}\{p_{\mathcal{K}}(c+\mathcal{A}^* y)\} + b\,, \end{aligned}$$

which is the required result. ∎

The dual function has a strong structure which will be used for algorithmic perspectives. The dual problem reduces to the *convex-differentiable* optimization problem

$$\begin{cases} \inf\ \tfrac{1}{2}||p_{\mathcal{K}}(c+\mathcal{A}^* y)||^2 - y^\top b, \\ y \in \mathbb{R}^m\,. \end{cases}$$

Example 1 (semidefinite least-squares). In the case $\mathcal{K} = \mathcal{S}_n^+$, the key point is that we have an easy-to-compute formulation of the projection (Theorem 2.4). The dual problem is here

$$\begin{cases} \min\ ||(C+\mathcal{A}^* y)_+||^2 - b^\top y \\ y \in \mathbb{R}^m\,. \end{cases}$$

4. A dual algorithm. We want to solve the primal problem, i.e., to find $x^\star \in \mathcal{K}$ closest to $c \in \mathbb{R}^p$ while satisfying affine constraints. The structure of the primal is not easy to use directly. On the other hand, its dual problem is more strongly structured (Theorem 3.2) and thus opens the way to a possible resolution procedure.

We assume in this section that there is a solution $y^\star$ to the dual problem: the dual function is bounded from above and its supremum is actually a maximum, achieved at $y^\star$. We are therefore in the following primal-dual situation

$$(4.1) \qquad (primal)\ \begin{cases} \min \quad \frac{1}{2}||x-c||^2 \\ x \in \mathcal{K}, \quad \mathcal{A}\,x = b \end{cases} \qquad (dual)\ \begin{cases} \max \quad \theta(y) \\ y \in \mathbb{R}^m\,. \end{cases}$$

with θ expressed by (3.3).

4.1. From dual to primal solution. In this subsection, we suppose that we are able to get efficiently $y^\star$. We show that in this case the primal problem is indeed solved. We start by mentioning that each value of the dual function gives a lower bound on the primal objective function: from the weak duality theorem (see [HUL93, Chap. XII]), there holds

$$(4.2) \qquad \theta(y) \leq \frac{1}{2}||c-x||^2$$

for all dual-feasible points (i.e., $y \in \mathbb{R}^m$) and for all primal-feasible points (i.e., $x \in \mathcal{K}$ such that $\mathcal{A}\,x = b$).

THEOREM 4.1. *Assume the existence of a dual solution $y^\star$. Then the solution $x^\star$ of the primal problem is given by*

$$(4.3) \qquad x^\star = p_{\mathcal{K}}(c + \mathcal{A}^* y^\star)\,.$$

Proof. From Theorem 3.2, θ is concave and differentiable, then at $y^\star$ which achieves its maximum, its gradient is zero. By equations (3.4) and (3.6), this results in $\mathcal{A}\,x(y^\star) = b$, i.e., $x(y^\star)$ is primal-feasible. Then we have by (3.5)

$$(4.4) \qquad \theta(y^\star) = L(x(y^\star); y^\star) = \frac{1}{2}||c - x(y^\star)||^2.$$

By (4.2), $\theta(y^\star)$ is a lower bound of the objective function of the primal. Equation (4.4) means that this lower bound is reached at the primal-feasible $x(y^\star)$. Thus that point is the minimum and

$$x^\star = x(y^\star) = p_{\mathcal{K}}(c + \mathcal{A}^* y^\star)\,,$$

which ends the proof. ∎

This theorem says that there is *no duality gap* between the primal and the dual. This is expressed by equation (4.4): the values of the primal function at its minimum and of the dual at its maximum are the same.

A particular case yielding both existence of $y^\star$ and absence of a duality gap is the primal *Slater condition* (see [HUL93, Chap. XII]), expressing that feasibility of the primal constraints is preserved despite perturbations of b. It corresponds to the existence of a point *strictly feasible* of the primal: there exists x satisfying $\mathcal{A}\,x = b$ and lying in the interior of $\mathcal{K}$, assumed nonempty.

4.2. Computing a dual solution. The regularity properties of θ allow the use of any classical algorithm to minimize it; for instance, a quasi-Newton algorithm is considered as most efficient.

ALGORITHM 1. *Consider the pair of primal-dual problems* (4.1). *Let a black-box perform the following task:*

(i) *compute* $\mathcal{A}^* y$ *for given* $y \in \mathbb{R}^m$;
(ii) *compute* $\mathcal{A}\, x$ *for given* $x \in \mathbb{R}^p$;
(iii) *compute* $p_{\mathcal{K}}(z)$ *for given* $z \in \mathbb{R}^p$.

Use a quasi-Newton optimization code to maximize θ *on* $\mathbb{R}^m$. *With the help of the above black-box, this code generates a maximizing sequence* $(y_k)_k$ *together with the corresponding:*

(i) $x_k = p_{\mathcal{K}}(c + \mathcal{A}^* y_k)$;
(ii) $\nabla\theta(y_k) = -\mathcal{A}\, x_k + b$;
(iii) $\theta(y_k) = -\frac{1}{2}||x_k||^2 + y^\top b$.

To implement the above algorithm the only thing we basically need is to compute $p_{\mathcal{K}}$. In other words, the key point to solve our problem (i.e., to compute the projection onto the intersection of $\mathcal{K}$ with an affine hyperplane) is to know how to solve the problem without affine constraints (i.e., to compute the projection onto $\mathcal{K}$). This means the algorithm is efficient when the difficulty is due to the addition of affine constraints. For instance, this is the case for semidefinite least-squares, where we have the easy-to-compute expression (2.4) of the projection.

An instance of quasi-Newton known to be convergent when the objective function is convex and has a Lipschitz gradient is the so-called BFGS with Wolfe line-search (Theorem 4.9 of [BGLS03]). Here is a convergence result.

THEOREM 4.2. *Let* $\mathcal{A}$ *be surjective and the Slater assumption hold. Then Algorithm* 1 *gives an approximation of* $x^\star$: *for any* $\epsilon > 0$, *there is* k *such that* $||x_k - x^\star|| \leq \epsilon$.

Proof. From the Slater assumption and the surjectivity of $\mathcal{A}$, the dual optimal set is bounded [HUL93, Chap. VII], and then each level-set is bounded [HUL93, Chap. IV]. The sequence (y_k) is thus bounded. Take $\epsilon > 0$. Since θ is continuous on $\mathbb{R}^m$, there exists a dual solution $y^\star$ and k large enough such that $||y_k - y^\star|| \leq \epsilon / ||A^*||$. Now from Lipschitzian property of the projection we can write

$$||x^\star - x_k|| = ||p_{\mathcal{K}}(c + \mathcal{A}^* y^\star) - p_{\mathcal{K}}(c + \mathcal{A}^* y_k)|| \leq ||\mathcal{A}^*||\ ||y^\star - y_k|| \leq \epsilon\,.$$

This ends the proof. □

We mention that the so-called *limited memory* quasi-Newton method can also be used. It avoids the need to store an $m \times m$ matrix, thus accommodating very large values (see [BGLS03, sects. 1.2.2 and 6.3]).

Remark 4.3. To conclude this section, we mention two possible extensions.

(i) Observe that the dual of

$$\begin{cases} \min \quad \frac{1}{2}||x - c||^2, \\ x \in \mathcal{K}, \quad \mathcal{A}\, x \leq b, \end{cases}$$

is (by an easy adaptation of the proof of Theorem 3.1)

$$\begin{cases} \max \quad \theta(y), \\ y \in \mathbb{R}^m, \quad y_i \geq 0 \quad \text{for all } i = 1, \ldots, m. \end{cases}$$

Thus Algorithm 1 can solve such problems with inequality constraints, whenever the quasi-Newton algorithm accepts box-constraints.

(ii) We also add that this approach can even be used for problems when $\mathcal{K}$ is a general closed convex set. The method can be adapted to treat the projection on the intersection of a convex set $\mathcal{K}$ and an affine subspace, if one knows how to project onto $\mathcal{K}$. In fact, θ is always a concave differentiable function, whose gradient is

$$\nabla\theta(y) = -\mathcal{A}x(y) + b$$

with $x(y) = \operatorname{argmin}_{x\in\mathcal{K}} L(x; y)$. All these results come from [HUL01, Chap. D.4.4].

5. Semidefinite least-squares. In this section, we focus on the case where $\mathcal{K} = \mathcal{S}_n^+$. Semidefinite least-squares is a very important subclass of the general least-squares problem (1.1). For instance, the computation of a "good" approximation of the covariance matrix of n assets can be expressed as a semidefinite least-squares problem (see subsection 5.4). Recall that an optimization program is written under a semidefinite least-squares form if there are a matrix $C \in \mathcal{S}_n$, a vector $b \in \mathbb{R}^m$ and a linear operator $\mathcal{A} : \mathcal{S}_n \longrightarrow \mathbb{R}^m$ such that

$$\text{(sdls)} \quad \begin{cases} \min & \frac{1}{2}||X - C||^2, \\ \mathcal{A}\,X = b, \\ X \succeq 0\,. \end{cases}$$

If $\mathcal{A}$ is expressed via its m component functions, (sdls) can be formulated with the help of m symmetric matrices:

$$\begin{cases} \min & \frac{1}{2}||X - C||^2, \\ \langle A_i, X\rangle = b_i\,, & i = 1, \ldots, m, \\ X \succeq 0\,. \end{cases}$$

We present, in subsections 5.1 and 5.2, known methods to solve (sdls): interior points and alternative projections. We also give a dual interpretation of the latter. We then show that Algorithm 1 is a good alternative to these methods.

5.1. Semidefinite approach. A natural idea to attack (sdls) directly is to phrase it as a semidefinite program. The problem can actually be seen as a quadratic-semidefinite program and then efficient interior-points methods for SDP programming are available (see [Tod01] for a review).

The (nonlinear) objective function can actually be pushed into constraints:

$$\begin{cases} \min & t, \\ ||X - C|| \leq t, \\ \mathcal{A}\,X = b, \\ X \succeq 0, \end{cases} \tag{5.1}$$

a problem expressed as a quadratic-semidefinite program [BTN01]. Thus powerful interior-points methods solvers can be used. However the number of variables is $\mathcal{O}(n^2)$ and this approach is presented as impractical for large n in [Hig02, subsect. 3.3]. Tests that we ran with SEDUMI [Stu99] confirm this point.

However, it should be mentioned that putting (5.1) in SEDUMI format requires the introduction of artificial variables and constraints. Adapted interior-points variants may exist. Note that [Tak03] provides one for sparse matrices.

5.2. Alternating projections method. Another interesting method is proposed in [Hig02] to solve particular instances of semidefinite least-squares (and it could be easily generalized to any semidefinite least-squares). Here $\mathcal{A}$ is the diag-operator: we want to solve

$$\begin{cases} \min \quad \frac{1}{2}||X-C||^2\,, \\ X_{ii}=1\,, \quad i=1,\dots,n\,, \\ X \succeq 0\,. \end{cases} \tag{5.2}$$

Introducing the notation

$$\mathcal{U}=\{X\in\mathcal{S}_n : X_{ii}=1\}, \tag{5.3}$$

the idea of the so-called alternating projection method is to repeat the operation

$$X \leftarrow p_{\mathcal{S}_n^+}(p_{\mathcal{U}}(X)). \tag{5.4}$$

Besides, the so-called Dykstra's correction [Dyk83] is used in [Hig02]. All together the algorithm is as follows.

ALGORITHM 2 (Algorithm 3.3 of [Hig02]). *For $C\in\mathcal{S}_n$, this algorithm solves* (5.2):

$\Delta S_0=0, Y_0=C$
for $k=1,2,\dots$
  $R_k=Y_{k-1}-\Delta S_{k-1}$ *% Dykstra's correction*
  $X_k=p_{\mathcal{S}_n^+}(R_k)$
  $\Delta S_k=X_k-R_k$
  $Y_k=p_{\mathcal{U}}(X_k)$
end

The alternating projection method has actually a dual interpretation in this case. Theorem 5.1 below says that it is just the standard gradient optimization algorithm applied to the dual of (5.2), namely $y_{k+1}=y_k+\nabla\theta(y_k)$. In fact, recalling formula (3.6), the gradient algorithm can be expressed as follows.

ALGORITHM 3. *This algorithm maximizes $\theta(y)=-||p_{\mathcal{S}_n^+}(C+\mathcal{A}^*y)||^2+b^\top y$ on $\mathbb{R}^n$, by the gradient method (with constant stepsize equal to 1):*

$y_1=0$,
for $k=1,2,\dots$
  $\bar{X}_k=p_{\mathcal{S}_n^+}(C+\mathcal{A}^*y_k)$
  $y_{k+1}=y_k+(-\mathcal{A}\bar{X}_k+b)$
end

In view of (5.2), we have here $b=[1,\dots,1]^\top$, $\mathcal{A}=\mathrm{diag}$, $\mathcal{A}^*=\mathrm{Diag}$, and we observe that $\mathcal{A}\mathcal{A}^*=I_m$.

THEOREM 5.1. *The sequence $(\bar{X}_k)$ generated by Algorithm* 3 *is the same as (X_k) generated by Algorithm* 2.

Proof. First, observe that the projection of $X\in\mathcal{S}_n$ on $\mathcal{U}$ is

$$p_{\mathcal{U}}(X)=X-\mathcal{A}^*(\mathcal{A}X-b). \tag{5.5}$$

Let us prove by recurrence that

$$\bar{X}_k=X_k \;\text{ and }\; R_k=C+\mathcal{A}^*y_k \quad \text{for all } k\geq 0\,. \tag{5.6}$$

This is true for $k = 1$ since $R_1 = C$, $y_1 = 0$, and $\bar{X}_1 = X_1 = p_{\mathcal{S}_n^+}(C)$. Suppose now that it holds for k; then

$$\begin{aligned}
R_{k+1} &= Y_k - \Delta S_k && \text{[definition of } R_{k+1}\text{]} \\
&= Y_k - X_k + R_k && \text{[definition of } \Delta S_k\text{]} \\
&= -\mathcal{A}^*(\mathcal{A}X_k - b) + R_k && \text{[(5.5) and definition of } Y_k\text{]} \\
&= -\mathcal{A}^*(\mathcal{A}\bar{X}_k - b) + C + \mathcal{A}^* y_k && \text{[recurrence assumptions]} \\
&= C + \mathcal{A}^*(y_k - (\mathcal{A}\bar{X}_k - b)) \\
&= C + \mathcal{A}^* y_{k+1}\,. && \text{[definition of } y_{k+1}\text{]}
\end{aligned}$$

Hence $X_{k+1} = p_{\mathcal{S}_n^+}(R_{k+1}) = p_{\mathcal{S}_n^+}(c + \mathcal{A}^* y_{k+1}) = \bar{X}_{k+1}$, and the theorem is proved. □

As a result, the method of alternative projections (Algorithm 2) and our proposal (Algorithm 1) are well comparable: both are optimization algorithms to maximize the dual function, the former does this by the (simple) gradient method with constant step size while the latter uses the (sophisticated) quasi-Newton approach.

5.3. Numerical results. For illustration, Algorithm 1 has been applied to instances of (5.2). We ran three types of experiments:

(i) We solve (5.2) with random dense matrices C (random $C_{ij} \in [-1, 1]$ and $C_{ii} = 1$) of sizes from 100×100 to 3000×3000.

(ii) We take a matrix $X^\star$ in $\mathcal{U} \cap \mathcal{S}_n^+$ (where $\mathcal{U}$ is defined by (5.3)) of size 1000×1000 and we perturb it to create a matrix C such that $X^\star$ is the projection of C. We then test Algorithm 1 with this C.

(iii) We fix the size (500×500) and we take matrices with increasing entries on the diagonal.

The algorithm has been coded in Fortran and we use the LAPACK library for numerical algebra. Note that the computation of the projection onto $\mathcal{S}_n^+$ is nothing more than an eigensystem computation: we use symmetric QR algorithm of LAPACK. In our experiments, the stopping test is

$$\frac{1}{\sqrt{n}}||\nabla\theta(y_k)|| = \frac{1}{\sqrt{n}}||\mathcal{A}X_k - b|| \leq 10^{-7}\,.$$

The performance measures have been obtained on a machine of the Intel P4 2 GHz processor family with 512 Mbytes of memory. The system runs under Linux Redhat 8.0 and uses the gnu compilation chain.

First experiment. The results with random matrices are as follows.

matrix sizes	cpu time	nb of iterations
100×100	0.2 s	14
300×300	3.3 s	14
500×500	16.3 s	17
800×800	1 min 10 s	17
1000×1000	2 min 05 s	18
1500×1500	7 min 35 s	18
2000×2000	17 min 41 s	19
3000×3000	1h 08 min 14 s	19

Some observations are worth mentioning:

- Computation on matrices up to 200×200 takes less than one second and the algorithm copes very well with larger matrices (one hour for a 3000×3000 dense matrix).

- For these kinds of matrices (with $C_{ij} \in [-1, 1]$), the typical number of iterations ranges between 10 and 20, almost independently of the problem size. The experimental computational cost is $\mathcal{O}(n^3)$.
- The bulk of the work is the spectral decomposition. Better cpu time could be obtained along the lines of Corollary 3.5 of [Hig02], by avoiding the computation of the full eigensystem. We have not implemented this idea.

Second experiment. Now we give an idea of the behavior of the algorithm by tracing the error between the exact solution and the current iterate. To construct a synthetic example, one can proceed as follows. Take a matrix $X^\star$ in $\mathcal{U} \cap \mathcal{S}_n^+$. Then set $C := X^\star + \delta X$, where δX lies in the normal cone to $\mathcal{U} \cap \mathcal{S}_n^+$ at $X^\star$, so that $X^\star$ solves (5.2).

By calculus rules of [HUL01, Chap. A], the normal cone to $\mathcal{U} \cap \mathcal{S}_n^+$ at $X^\star$ is the sum of $\mathcal{U}^\perp$ (normal "cone" to the subspace $\mathcal{U}$) and of $\mathcal{S}_n^- \cap X^\perp$ (normal cone to $\mathcal{S}_n^+$ at $X^\star$). It suffices to choose an $X^\star$ such that constructing a matrix in $\mathcal{S}_n^- \cap X^\perp$ is easy. For example, take $1 \leq \ell \leq n$ and consider the matrix

$$X^\star := \begin{bmatrix} E_\ell & \\ & I_{n-\ell} \end{bmatrix},$$

where $I_{n-\ell}$ is the $(n-\ell)\times(n-\ell)$ identity matrix and E_ℓ is the $\ell \times \ell$ matrix with all entries equal to 1. Then it is easy to see that a suitable matrix is

$$C := \begin{bmatrix} \frac{\ell}{\ell-1} E_\ell & \\ & I_{n-\ell} \end{bmatrix} + D,$$

where D is an arbitrary diagonal matrix.

Figure 5.1 shows the evolution of the distance of the current iterate X_k to the solution (which is $X^\star$ by construction) for an instance where $n = 1000$, $\ell = 500$, and D_{ii} is a random number in $[-10, 10]$. The algorithm converges in 56 iterations (and 58 diagonalizations). Note that the run takes more iterations than the 1000×1000-instance of the first experiment (18 iterations). This fact is underlined by the third experiment.

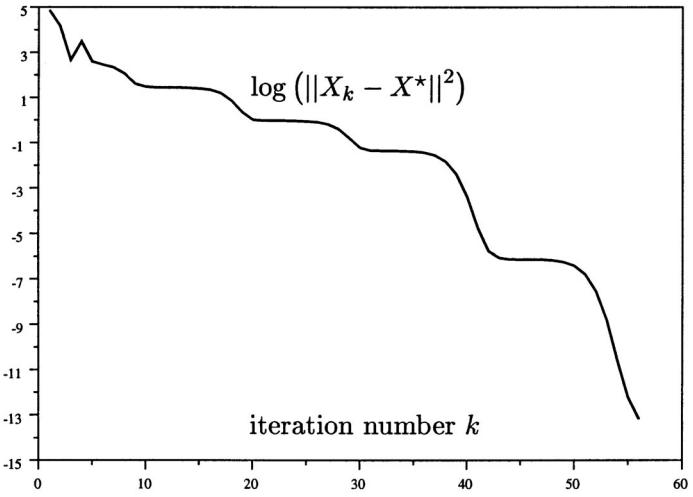


FIG. 5.1. *Evolution of* $||X_k - X^\star||^2$.

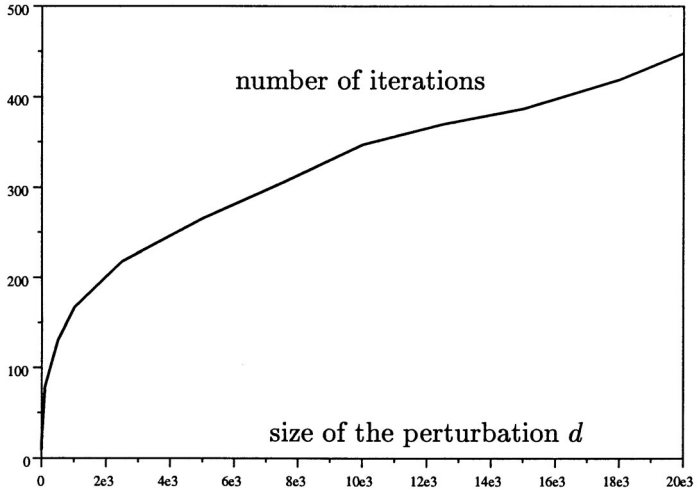


FIG. 5.2. *Influence of the remoteness of C.*

Third experiment. When the entries of C get larger, C gets more remote from the target $\mathcal{U} \cap \mathcal{S}_n^+$. One can think that computing the projection of C then takes more iterations. The aim of this experiment is to quantify the phenomenon, which turns out to be fairly significant. In Figure 5.2, C is constructed as in the second experiment with $n = 500$, $\ell = 250$, and random $D_{ii} \in [-d, d]$ (we increase d from 0 to 20000).

5.4. Calibration of covariance matrices. In this subsection we outline the problem of finding a "good" approximation $\bar{Q}$ of the theoretical covariance matrix Q between n assets: it turns out to be a semidefinite least-squares problem. Another problem of that kind is computing the nearest correlation matrix [Hig02]. To have a good approximation $\bar{Q}$ is important in portfolio management: this matrix is used to have a robust estimation of the "ex-ante" risk of any possible portfolio among these n assets.

For instance, portfolio managers often look for portfolios minimizing the financial risk while having a fixed return. They want to solve a portfolio selection problem of the following type:

$$\begin{cases} \min \quad x^\top Q x\,, \\ x^\top r \geq \beta\,, \\ x_i \in [0,1], \quad \sum_{i=1}^n x_i = 1. \end{cases} \tag{5.7}$$

This is the famous portfolio selection problem of Markowitz (Nobel prize winner in 1990). The covariance matrix Q (which is positive semidefinite) is used to estimate the risk. Under the classical economic assumption that there is no rewarding risk-less investments (no-arbitrage assumption), Q is definite positive. Let $\tilde{Q}$ be a first estimate of the true covariance matrix Q: for instance $\tilde{Q}$ can be the empirical estimate after k days. The point is that $\tilde{Q}$ has a bad condition number:

- When the number of observations k is too small, $\tilde{Q}$ is rank-deficient (some investments are considered with no risk; it is not consistent with the no-arbitrage assumption).
- When there are different levels of risks in the portfolio, $\tilde{Q}$ is ill conditioned (the condition number of $\tilde{Q}$ is typically greater than 10^7 if there are stocks,

options, and monetary products in the portfolio; it reaches 10^{17} for hedge funds).

We may impose $X \succeq \alpha I_n$ for some selected $\alpha > 0$ to avoid too low-risk portfolios: we thus guarantee "cautious" risk evaluations, and stability of the portfolio selected by (5.7). Eventually we are led to the so-called *calibration of covariance matrix* problem, which is a shifted (sdls):

$$\begin{cases} \min \quad \frac{1}{2}||X - \tilde{Q}||^2, \\ X \succeq \alpha I_n\,, \\ \langle I_n, X\rangle = \mathrm{tr}(\tilde{Q})\,, \\ \langle A_i, X\rangle = \sigma_i^2\,, \end{cases}$$

where σ_i^2 represent "ex-post" volatilities of well-chosen portfolios. The constraint $\langle I_n, X\rangle = \mathrm{tr}(\tilde{Q})$ enforces the conservation of the empirical total risk. We solve real-life instances of this problem (provided to us by RAISE PARTNER) with Algorithm 1; the results are quite similar to those of subsection 5.3.

We end with a remark. The material developed in this section can be easily extended to the Frobenius norm with weights (see Remark 2.5). This is of little impact in theory but more in practice: for instance, the covariance between some assets are sometimes more relevant than others, so we want to ensure in the calibration process of the covariance matrix that the relevance is properly emphasized.

Acknowledgments. My primary thanks go to Claude Lemaréchal for his advice and his everyday help. I am indebted to François Oustry and RAISE PARTNER (especially Nabil Layaïda). I also want to thank the editor and the two referees for their numerous suggestions.

REFERENCES

[BGLS03] J. F. Bonnans, J. Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal, *Numerical Optimization*, Springer-Verlag, Berlin, 2003.

[BTN01] R. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*, MPS-SIAM Ser. Optim., SIAM, Philadelphia, 2001.

[DH00] P. Davies and N. Higham, *Numerically stable generation of correlation matrices and theirs factors*, BIT, 40 (2000), pp. 640–651.

[Dyk83] R. L. Dykstra, *An algorithm for restricted least squares regression*, J. Amer. Statist. Assoc., 78 (1983), pp. 837–842.

[Hig88] N. Higham, *Computing a nearest symmetric positive semidefinite matrix*, Linear Algebra Appl., 103 (1988), pp. 103–118.

[Hig02] N. Higham, *Computing the nearest symmetric correlation matrix—a problem from finance*, IMA J. Numer. Anal., 22 (2002), pp. 329–343.

[HUL93] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, 1993.

[HUL01] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*, Springer-Verlag, Berlin, 2001.

[Stu99] J. F. Sturm, *Using Sedumi* 1.02, *a Matlab toolbox for optimization over symmetric cones*, Optm. Meth. Soft., 11/12 (1999), pp. 625–653.

[Tak03] P. L. Takouda, *Problèmes d'approximation matricielle linéaires coniques: Approches par projections et via Optimisation sous contraintes de semidéfinie positivité*, Ph.D. thesis, Université Paul Sabatier–Toulouse III, 2003.

[Tod01] M. J. Todd, *Semidefinite optimization*, Acta Numer., 10 (2001), pp. 515–560.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 1, pp. 285–294

ON SOME INVERSE EIGENVALUE PROBLEMS WITH TOEPLITZ-RELATED STRUCTURE*

FASMA DIELE†, TERESA LAUDADIO‡, AND NICOLA MASTRONARDI†

Abstract. Some inverse eigenvalue problems for matrices with Toeplitz-related structure are considered in this paper. In particular, the solutions of the inverse eigenvalue problems for Toeplitz-plus-Hankel matrices and for Toeplitz matrices having all double eigenvalues are characterized, respectively, in close form. Being centrosymmetric itself, the Toeplitz-plus-Hankel solution can be used as an initial value in a continuation method to solve the more difficult inverse eigenvalue problem for symmetric Toeplitz matrices. Numerical testing results show a clear advantage of such an application.

Key words. Toeplitz matrix, Toeplitz-plus-Hankel matrix, discrete cosine transform, eigenvalues, inverse eigenvalue problem, centro-symmetric isospectral flows.

AMS subject classifications. 65F15, 65F18

DOI. 10.1137/S0895479803430680

1. Introduction.

An inverse eigenvalue problem concerns the reconstruction of a matrix from assigned spectral data. This inverse problem arises in a remarkable variety of applications ranging from applied mechanics and physics to numerical analysis. See [6], which includes an extensive survey of such structured problems, including Jacobi, Toeplitz, nonnegative, and stochastic inverse problems.

Two types of inverse eigenvalues problems are considered in this paper. The first one concerns the construction of a Toeplitz-plus-Hankel matrix with prescribed spectrum. The second one concerns the construction of a Toeplitz matrix with double eigenvalues. Both problems are solved in closed form.

The first problem has already been studied in [1], where it was shown that, given n real values,

$$\lambda_1, \lambda_2, \dots, \lambda_n, \tag{1.1}$$

there are exactly $n!$ different τ-class matrices, i.e., symmetric and centrosymmetric matrices with Toeplitz-plus-Hankel structure with (1.1) as eigenvalues. In this paper, by exploiting the properties of the Chebyshev polynomials [27], we first construct n idempotent rank-one Toeplitz-plus-Hankel matrices $C_k^{(n)}$, $k = 0, 1, \dots, n-1$, such that ${C_k^{(n)}}^T C_j^{(n)} = 0$, $k \neq j$, where 0 is the zero matrix of order n. Hence, given the n values (1.1), any linear combination

$$\sum_{k=1}^{n} \lambda_{\Pi_{(k)}} C_{k-1}^{(n)}, \tag{1.2}$$

where $\Pi(\cdot)$ is any one of the $n!$ permutations of the set of indexes $\{1, 2, \dots, n\}$, solves the Toeplitz-plus-Hankel inverse eigenvalue problem (TpHIEP). Such a construction

*Received by the editors June 24, 2003; accepted for publication (in revised form) December 2, 2003; published electronically September 14, 2004.
http://www.siam.org/journals/simax/26-1/43068.html

†Istituto per le Applicazioni del Calcolo "M. Picone," Sez. Bari, via Amendola 122/D, I–70126 Bari, Italy (f.diele@area.ba.cnr.it, n.mastronardi@area.ba.cnr.it). This work was partially supported by MIUR grant number 2002014121.

‡Department of Electrical Engineering, ESAT-SISTA/COSIC, Katholieke Universiteit Leuven, Kardinaal Mercierlaan 94, 3001 Heverlee, Belgium (Teresa.Laudadio@esat.kuleuven.ac.be).

can be used to undertake the second problem of building Toeplitz matrices whose eigenvalues have multiplicity 2.

The question of solvability of Toeplitz inverse eigenvalue problem (TIEP) with distinct eigenvalues has been addressed by Landau [23], yielding the proof of the existence of a (*regular*) Toeplitz matrix with distinct eigenvalues. Unfortunately, his proof is not constructive. Alternatively, iterative schemes such as Newton's method [3] or algebraic procedures [24, 29] have been successfully introduced to solve numerically the problem. More recently, a continuation technique based on the solution of an *isospectral flow* has been proposed in [4] for solving TIEP. The flow evolves in the space of symmetric and centrosymmetric matrices and has Toeplitz matrices as equilibria. Although there is no rigorous proof, methods based on this idea seems to converge numerically to a regular Toeplitz matrix if a symmetric centrosymmetric matrix is used as the initial value. In this paper, we also examine the dynamical behavior of the flow approach when (1.2) is used as the initial value.

The paper is organized as follows. In section 2, we briefly review some properties of the Chebyshev polynomial of first kind, which will be used to determine the solutions of TpHIEP. Taking into account recent results on the distribution of the eigenvalues of symmetric Toeplitz matrices [29], the matrices (1.2) are imbedded in section 3 into matrices of double size in order to construct a solution for the TIEP where all eigenvalues have double multiplicity. In section 4 we study the convergence behavior of isospectral flow described in [12] if the matrix (1.2) is used as an initial value, followed by the conclusions.

2. Inverse eigenvalue problem for Toeplitz-plus-Hankel matrices. In this section we consider the inverse eigenvalue problem for Toeplitz-plus-Hankel matrices.

In particular, given the n values (1.1), known $\lambda_1, \lambda_2, \dots, \lambda_n$, we show how to construct $n!$ symmetric Toeplitz-plus-Hankel matrices, with eigenvalues (1.1).

We introduce the Chebyshev polynomials of first kind,

$$P_k(x) = \cos k\theta,\ x = \cos\theta, k \in \mathbb{N},$$

and the *orthonormalized* Chebyshev polynomials,

$$\tilde{P}_k(x) = \gamma_k P_k(x),\ \text{with } \gamma_k = \begin{cases} \sqrt{\frac{1}{\pi}} & \text{if } k = 0, \\ \sqrt{\frac{2}{\pi}} & \text{if } k > 0. \end{cases}$$

The zeros of $\tilde{P}_k,\ k > 0$, are given by

$$\zeta_j^{(k)} = \cos\theta_j^{(k)},\ \ \theta_j^{(k)} = \frac{2j-1}{k}\frac{\pi}{2},\ j = 1, \dots, k. \tag{2.1}$$

The polynomials $\tilde{P}_k$ satisfy the orthogonality relationship

$$\int_{-1}^{1} \tilde{P}_j(x)\tilde{P}_k(x)\frac{dx}{\sqrt{1-x^2}} = \frac{\pi}{n}\sum_{i=1}^{n} \tilde{P}_j(\zeta_i^{(n)})\tilde{P}_k(\zeta_i^{(n)}) = \delta_{j,k}, \quad k, j < n. \tag{2.2}$$

From (2.2), it follows that the so-called *Chebyshev–Vandermonde* matrices

$$\mathbf{V}_n = \sqrt{\frac{\pi}{n}}\begin{bmatrix} \tilde{P}_0(\zeta_1^{(n)}) & \tilde{P}_1(\zeta_1^{(n)}) & \dots & \tilde{P}_{n-1}(\zeta_1^{(n)}) \\ \tilde{P}_0(\zeta_2^{(n)}) & \tilde{P}_1(\zeta_2^{(n)}) & \dots & \tilde{P}_{n-1}(\zeta_2^{(n)}) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{P}_0(\zeta_n^{(n)}) & \tilde{P}_1(\zeta_n^{(n)}) & \dots & \tilde{P}_{n-1}(\zeta_n^{(n)}) \end{bmatrix}, \quad n \in \mathbb{N}, \tag{2.3}$$

are orthogonal. We remark that the matrices $\mathbf{V}_n$, $n \in \mathbb{N}$ are better known as the *discrete cosine transform* II of order n [26, 31].

PROPOSITION 2.1. *For $k = 0, \ldots, n-1$, define*

$$\mathbf{q}_k^{(n)} \stackrel{\text{def}}{=} \mathbf{V}_n(:, k+1).$$

Then the symmetric idempotent rank-one matrices

$$\mathbf{C}_k^{(n)} = \mathbf{q}_k^{(n)} (\mathbf{q}_k^{(n)})^T, \quad k = 0, \ldots, n-1,$$

are Toeplitz-plus-Hankel.

Proof. Let $\alpha_j = \sqrt{\frac{\pi}{n}}\gamma_j$, $j = 0, \ldots, n-1$. Since

$$\cos\alpha \cos\beta = \frac{1}{2}\left(\cos(\alpha+\beta) + \cos(\alpha-\beta)\right), \tag{2.4}$$

the (i,j) entry of $\mathbf{C}_k^{(n)}$ is given by

$$\begin{aligned}
\mathbf{C}_k^{(n)}(i,j) &= \frac{\alpha_k^2}{2} \cos\left(k\frac{2i-1}{n}\frac{\pi}{2}\right) \cos\left(k\frac{2j-1}{n}\frac{\pi}{2}\right) \\
&= \frac{\alpha_k^2}{2}\left(\cos\left(k\frac{2i-1-(2j-1)}{n}\frac{\pi}{2}\right) + \cos\left(k\frac{2i-1+2j-1}{n}\frac{\pi}{2}\right)\right) \\
&= \frac{\alpha_k^2}{2}\left(\cos\frac{k|i-j|\pi}{n} + \cos\frac{k(i+j-1)\pi}{n}\right).
\end{aligned}$$

Write

$$\mathbf{T}_k^{(n)} \stackrel{\text{def}}{=} \frac{\alpha_k^2}{2}\begin{bmatrix}
1 & \cos\frac{k\pi}{n} & \ddots & \cos\frac{k(n-2)\pi}{n} & \cos\frac{k(n-1)\pi}{n} \\
\cos\frac{k\pi}{n} & 1 & \cos\frac{k\pi}{n} & \ddots & \cos\frac{k(n-2)\pi}{n} \\
\ddots & \ddots & \ddots & \ddots & \ddots \\
\cos\frac{k(n-2)\pi}{n} & \ddots & \ddots & 1 & \cos\frac{k\pi}{n} \\
\cos\frac{k(n-1)\pi}{n} & \cos\frac{k(n-2)\pi}{n} & \ddots & \cos\frac{k\pi}{n} & 1
\end{bmatrix}$$

and

$$\begin{aligned}
\mathbf{H}_k^{(n)} &\stackrel{\text{def}}{=} \frac{\alpha_k^2}{2}\begin{bmatrix}
\cos\frac{k\pi}{n} & \cos\frac{k2\pi}{n} & \iddots & \cos\frac{k(n-1)\pi}{n} & \cos\frac{kn\pi}{n} \\
\cos\frac{k2\pi}{n} & \iddots & \iddots & \cos\frac{kn\pi}{n} & \cos\frac{k(n+1)\pi}{n} \\
\iddots & \iddots & \iddots & \iddots & \iddots \\
\cos\frac{k(n-1)\pi}{n} & \cos\frac{kn\pi}{n} & \iddots & \iddots & \cos\frac{k(2n-2)\pi}{n} \\
\cos\frac{kn\pi}{n} & \cos\frac{k(n+1)\pi}{n} & \iddots & \cos\frac{k(2n-2)\pi}{n} & \cos\frac{k(2n-1)\pi}{n}
\end{bmatrix} \\
&= \frac{\alpha_k^2}{2}\begin{bmatrix}
\cos\frac{k\pi}{n} & \cos\frac{k2\pi}{n} & \iddots & \cos\frac{k(n-1)\pi}{n} & (-1)^k \\
\cos\frac{k2\pi}{n} & \iddots & \iddots & (-1)^k & \cos\frac{k(n-1)\pi}{n} \\
\iddots & \iddots & \iddots & \iddots & \iddots \\
\cos\frac{k(n-1)\pi}{n} & (-1)^k & \iddots & \iddots & \cos\frac{k2\pi}{n} \\
(-1)^k & \cos\frac{k(n-1)\pi}{n} & \iddots & \cos\frac{k2\pi}{n} & \cos\frac{k\pi}{n}
\end{bmatrix}.
\end{aligned} \tag{2.5}$$

Then it follows that $\mathbf{C}_k^{(n)} = \mathbf{T}_k^{(n)} + \mathbf{H}_k^{(n)}$. ∎

We remark that columns of the orthogonal matrix

$$\mathbf{W}_n = \left[\sqrt{\frac{2}{n}}\left(\cos\frac{(i-1/2)(2j-1)\pi}{2n+1}\right)\right]_{i,j=1}^{n},$$

related to the Chebyshev polynomial of the third kind, can be used to solve the TpHIEP.

PROPOSITION 2.2. *The matrix*

$$\mathbf{U}_n = \sqrt{\frac{2}{n}}\begin{bmatrix} \sin(\theta_1^{(n)}) & \sin 2(\theta_1^{(n)}) & \dots & \sqrt{\frac{1}{2}}\sin n(\theta_1^{(n)}) \\ \sin(\theta_2^{(n)}) & \sin 2(\theta_2^{(n)}) & \dots & \sqrt{\frac{1}{2}}\sin n(\theta_2^{(n)}) \\ \vdots & \vdots & \ddots & \vdots \\ \sin(\theta_n^{(n)}) & \sin 2(\theta_n^{(n)}) & \dots & \sqrt{\frac{1}{2}}\sin n(\theta_n^{(n)}) \end{bmatrix}, \quad n \in \mathbb{N}, \tag{2.6}$$

is orthogonal, with $\theta_j^{(n)}$ *defined in* (2.1). *Moreover, let*

$$\mathbf{u}_k^{(n)} \stackrel{\text{def}}{=} \mathbf{U}_n(:,k), \quad k = 1, \dots, n.$$

Then the symmetric rank-one matrices

$$\mathbf{G}_k^{(n)} = \mathbf{u}_k^{(n)}(\mathbf{u}_k^{(n)})^T, \quad k = 1, \dots, n,$$

are Toeplitz-minus-Hankel.

Proof. The matrix $\mathbf{U}_n$ is orthogonal because the following relation holds:

$$\mathbf{U}_n = \Sigma_n \mathbf{V}_n \mathbf{J}_n,$$

where

$$\mathbf{J}_n = \begin{bmatrix} 0 & & 1 \\ & \iddots & \\ 1 & & 0 \end{bmatrix} \quad \text{and} \quad \Sigma_n = \operatorname{diag}(1, -1, \dots, (-1)^{i+1}, \dots, (-1)^{n+1}).$$

Furthermore, the matrices $\mathbf{G}_k^{(n)}$ are Toeplitz-minus-Hankel because

$$\sin\alpha \sin\beta = \frac{1}{2}\left(\cos(\alpha-\beta) - \cos(\alpha+\beta)\right). \qquad \square$$

We recall that a vector $\mathbf{x} \in \mathbb{R}^n$ is *symmetric* if $\mathbf{J}_n\mathbf{x} = \mathbf{x}$ and $\mathbf{x}$ is *skew-symmetric* if $\mathbf{J}_n\mathbf{x} = -\mathbf{x}$. Thus we note that $\mathbf{q}_k^{(n)}$ is symmetric when k is even and skew-symmetric when k is odd; whereas $\mathbf{u}_k^{(n)}$ is symmetric when k is odd and skew-symmetric when k is even.

The inverse eigenvalue problem for matrices with Toeplitz-plus-Hankel structure can now be solved in closed form as follows.

PROPOSITION 2.3. *Given* $\lambda_i \in \mathbb{R}$, $i = 1, \dots, n$, *the matrix*

$$\mathbf{A}_\mathbf{C} = \sum_{k=0}^{n-1} \lambda_{k+1} \mathbf{C}_k^{(n)} \tag{2.7}$$

is Toeplitz-plus-Hankel. Moreover,

$$\mathbf{A_C}\mathbf{q}_{k-1}^{(n)} = \lambda_k \mathbf{q}_{k-1}^{(n)}, \tag{2.8}$$

i.e., λ_k*, for each* $k = 1, \ldots, n$*, is an eigenvalue of* $\mathbf{A_C}$ *with corresponding eigenvector* $\mathbf{q}_{k-1}^{(n)}$.

Proof. The matrix $\mathbf{A_C}$ is Toeplitz-plus-Hankel because it is the sum of the Toeplitz-plus-Hankel matrices $\mathbf{C}_k^{(n)}$, $k = 0, \ldots, n-1$. Moreover

$$\mathbf{A_C}\mathbf{q}_{k-1}^{(n)} = \sum_{j=0}^{n-1} \lambda_{j+1}\mathbf{C}_j^{(n)}\mathbf{q}_{k-1}^{(n)} = \sum_{j=0}^{n-1} \lambda_{j+1}\left((\mathbf{q}_j^{(n)})^T\mathbf{q}_{k-1}^{(n)}\right)\mathbf{q}_j^{(n)} = \lambda_k\mathbf{q}_{k-1}^{(n)}$$

since $\mathbf{q}_j^{(n)}$, $j = 0, \ldots, n-1$, are the columns of the orthogonal Chebyshev–Vandermonde matrix (2.6). ∎

PROPOSITION 2.4. *Given* $\lambda_i \in \mathbb{R}$, $i = 1, \ldots, n$*, the matrix*

$$\mathbf{A_G} = \sum_{k=1}^{n} \lambda_k \mathbf{G}_k^{(n)} \tag{2.9}$$

is Toeplitz-minus-Hankel. Moreover,

$$\mathbf{A_G}\mathbf{u}_k^{(n)} = \lambda_k \mathbf{u}_k^{(n)}, \tag{2.10}$$

i.e., λ_k*, for each* $k = 1, \ldots, n$*, is an eigenvalue of* $\mathbf{A_G}$ *with corresponding eigenvector* $\mathbf{u}_k^{(n)}$.

We note that the matrix (2.7) or (2.9) is but one of the $n!$ possible choices. In fact, any matrix of the form $\sum_{k=1}^{n} \lambda_{\Pi(k)}\mathbf{C}_{k-1}^{(n)}$ or $\sum_{k=1}^{n} \lambda_{\Pi(k)}\mathbf{G}_{k-1}^{(n)}$ solves the TpHIEP, where $\Pi(\cdot)$ is any one of the $n!$ permutations of the set of indexes $\{1, 2, \ldots, n\}$.

We observe further that, for any $\lambda_1, \ldots, \lambda_n$, the matrices $\mathbf{A_C}$ and $\mathbf{A_G}$ are also centrosymmetric. So the inverse eigenvalue problem for centrosymmetric matrices is solved as well in our context.

3. Inverse Toeplitz eigenvalue problem for eigenvalues having double multiplicity. In this section we show how to construct a Toeplitz matrix with prescribed double eigenvalues. Our idea is to imbed the matrices (2.7) and (2.9) into a Toeplitz matrix of double size.

Before doing it, let us introduce the following results [29]. Let $(t_0, t_1, \ldots, t_{n-1})^T$ a vector and let $\mathbf{T_n} \stackrel{\text{def}}{=} \left(t_{|i-j|}\right)_{i,j=1}^{n}$ an n-dimensional Toeplitz symmetric matrix.

THEOREM 3.1. *Suppose that* $n = 2m$ *and* μ *is an eigenvalue of*

$$\mathbf{A} \stackrel{\text{def}}{=} \left[t_{|i-j|} + t_{n-i-j+1}\right]_{i,j=1}^{m}$$

with associated unit eigenvector $\mathbf{x}$*. Then* μ *is an even eigenvalue of* $\mathbf{T_n}$*, with associated symmetric unit eigenvector*

$$\mathbf{p} = \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{J}_m\mathbf{x} \\ \mathbf{x} \end{bmatrix}.$$

THEOREM 3.2. *Suppose that* $n = 2m$ *and* μ *is an eigenvalue of*

$$\mathbf{B} = \left[t_{|i-j|} - t_{n-i-j+1}\right]_{i,j=1}^{m},$$

with associated unit eigenvector $\mathbf{y}$. *Then* μ *is an odd eigenvalue of* $\mathbf{T_n}$ *with associated skew-symmetric unit eigenvector*

$$\mathbf{q} = \frac{1}{\sqrt{2}} \begin{bmatrix} -\mathbf{J}_m\mathbf{y} \\ \mathbf{y} \end{bmatrix}.$$

Given n real numbers $\lambda_i \in \mathbb{R}$, $i = 1, \ldots, n$, we can now solve the following inverse Toeplitz eigenvalue problem: *Given* n *real numbers* $\lambda_i \in \mathbb{R}$, $i = 1, \ldots, n$, *find a symmetric Toeplitz matrix* $\mathbf{T_{2n}}$ *of order* $2n$ *such that* λ_i, $i = 1, \ldots, n$, *are double eigenvalues of* $\mathbf{T_{2n}}$.

This inverse eigenvalue problem represents a rare but challenging scenario for Toeplitz matrices. Existing numerical methods such as the Newton iteration are known to fail unless the parity is taken into account by evenly assigning half of the spectrum into even and the other half into odd eigenvalues. Here we face the problem by exploiting the previous results on TpHIEP.

Let $\mathbf{u}_0^{(n)} \stackrel{\text{def}}{=} [\mathbf{U}_n(1:n,n)]$, $\tilde{\mathbf{U}}_n \stackrel{\text{def}}{=} [\mathbf{u}_0^{(n)}, \mathbf{u}_1^{(n)}, \ldots, \mathbf{u}_{n-1}^{(n)}]$, and $\Lambda_n = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$. Let us consider

$$\mathbf{A_C} = \sum_{k=0}^{n-1} \lambda_{k+1}\mathbf{T}_k^{(n)} + \sum_{k=0}^{n-1} \lambda_{k+1}\mathbf{H}_k^{(n)} \stackrel{\text{def}}{=} \mathbf{T} + \mathbf{H}.$$

Let

$$\tilde{\mathbf{A}}_\mathbf{G} \stackrel{\text{def}}{=} \mathbf{T} - \mathbf{H} = \sum_{k=0}^{n-1} \lambda_{k+1}\mathbf{T}_k^{(n)} - \sum_{k=0}^{n-1} \lambda_{k+1}\mathbf{H}_k^{(n)} = \sum_{k=0}^{n-1} \lambda_{k+1}\mathbf{u}_k^{(n)}(\mathbf{u}_k^{(n)})^T.$$

Let

$$\mathbf{T_{2n}} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{T} & \mathbf{JH} \\ (\mathbf{JH})^T & \mathbf{T} \end{bmatrix}.$$

Applying Theorem 3.1, we see that

$$\frac{1}{\sqrt{2}}\mathbf{T_{2n}} \begin{bmatrix} \mathbf{J}_n\mathbf{V}_n \\ \mathbf{V}_n \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{J}_n\mathbf{V}_n \\ \mathbf{V}_n \end{bmatrix} \Lambda_n.$$

Moreover, applying Theorem 3.2, we see that

$$\frac{1}{\sqrt{2}}\mathbf{T_{2n}} \begin{bmatrix} -\mathbf{J}_n\tilde{\mathbf{U}}_n \\ \tilde{\mathbf{U}}_n \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} -\mathbf{J}_n\tilde{\mathbf{U}}_n \\ \tilde{\mathbf{U}}_n \end{bmatrix} \Lambda_n.$$

Thus, if

$$\mathbf{Q}_{2n} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{J}_n\mathbf{V}_n & -\mathbf{J}_n\tilde{\mathbf{U}}_n \\ \mathbf{V}_n & \tilde{\mathbf{U}}_n \end{bmatrix} \text{ and } \Delta_{2n} = \begin{bmatrix} \Lambda_n & \\ & \Lambda_n \end{bmatrix},$$

then

$$\mathbf{T_{2n}} = \mathbf{Q}_{2n}\Delta_{2n}\mathbf{Q}_{2n}^T. \qquad \square$$

4. Initial guesses for continuation methods for TIEP. The solution to TpHIEP can help to solve the general TIEPs. We briefly recall some theoretical background on which continuation methods for TIEP given in [12] is based. The technique essentially amounts to numerically approximating the solution of the isospectral flow:

$$X'(t) = [k(X(t)), X(t)], \qquad t > 0, \qquad X(0) = X_0 = X_0^T, \tag{4.1}$$

where $k(X) = (k_{ij}(X))_{ij}^n$ is defined as

$$k_{i,j} = \begin{cases} x_{i+1,j} - x_{i,j-1} & \text{if} \quad 1 \le i < j \le n, \\ 0 & \text{if} \quad 1 \le i = j \le n, \\ x_{i-1,j} - x_{i,j+1} & \text{if} \quad 1 \le j < i \le n. \end{cases} \tag{4.2}$$

If the initial value X_0 is a symmetric and centrosymmetric matrix, the flow remains in the space of symmetric and centrosymmetric matrices where Toeplitz matrices are critical points of (4.1). In [12] it is shown that the convergence strongly depends on the choice of the starting matrix X_0. In fact, a condition for the convergence of the isospectal flow is that both the initial guess and matrix, a regular Toeplitz matrix, have the (same) eigenvalues alternating in parity when they are arranged in descending order.

The centrosymmetric Toeplitz-plus-Hankel matrices (2.7) and the centrosymmetric Toeplitz-minus-Hankel matrices (2.9) can be considered as a choice for the starting values of the isospectral flow. Given $\lambda_1 > \lambda_2 > \cdots > \lambda_n$ and taking into account parity features of the vectors $\mathbf{q}_k^{(n)}$ and $\mathbf{u}_k^{(n)}$, the matrices $\mathbf{A_C}$ and $\mathbf{A_G}$ in (2.7) and (2.9), automatically share the correct eigenvector-eigenvalue parity assignment, respectively.

5. Numerical tests. In this section we give some numerical results on the convergence behavior of the isospectral flow (4.1) if the matrices $\mathbf{A_C}$ and $\mathbf{A_G}$ in (2.7) and (2.9) are used as initial values. The performance is compared with the one when the Jacobi centrosymmetric matrix is considered as initial guesses. The fourth order method with step $h = 0.01$ described in [12] has been used. We will refer to equilibrium when the difference between two successive approximations of the solution of the flow (4.1) is less than $1e-8$. In all figures we report the history of the 2-norm of the annihilator $k(X(t))$ defined in (4.2). Since $k(X(t))$ vanishes when a matrix is a Toeplitz one, the evolution of its norm indicates how the flow tends towards a matrix with Toeplitz structure.

Example 1. Consider the case with eigenvalues $(\lambda_1, \lambda_1, \lambda_2, \lambda_2, \lambda_3, \lambda_3)$, where $\lambda_1 = 4 + \sqrt{11}$, $\lambda_2 = 3$, $\lambda_3 = 4 - \sqrt{11}$. In this case, the inverse eigenvalues problem is theoretically solved by the matrix with double eigenvalues predicted in section 3 and identified by its first row

$$(3.6667,\ 0.3861,\ 1.8250 - 0.7722,\ 1.8250,\ 0.3861).$$

Note that the isospectral flow (4.1) attains a different Toeplitz matrix with the first row given by

$$(3.6667,\ 1.9148,\ 0.3333,\ -0.0000,\ -0.3333,\ -1.9148).$$

In this case, the best choice for the initial value is $\mathbf{A_G}$, as shown in Figure 5.1 (left).

We will consider now eigenvalues with very different sizes:

$$(\lambda_1, \ldots, \lambda_6) = (0.01,\ 0.1,\ 1,\ 10,\ 100,\ 1000).$$

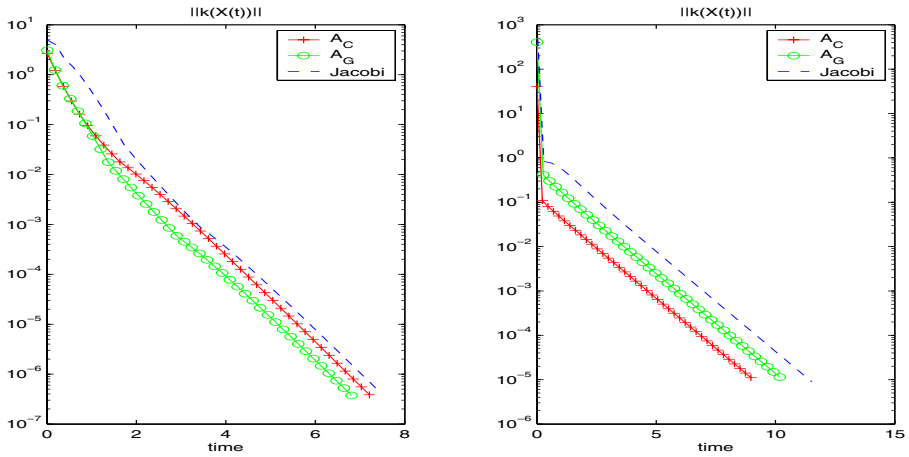


FIG. 5.1. *Convergence for* $n = 6$.

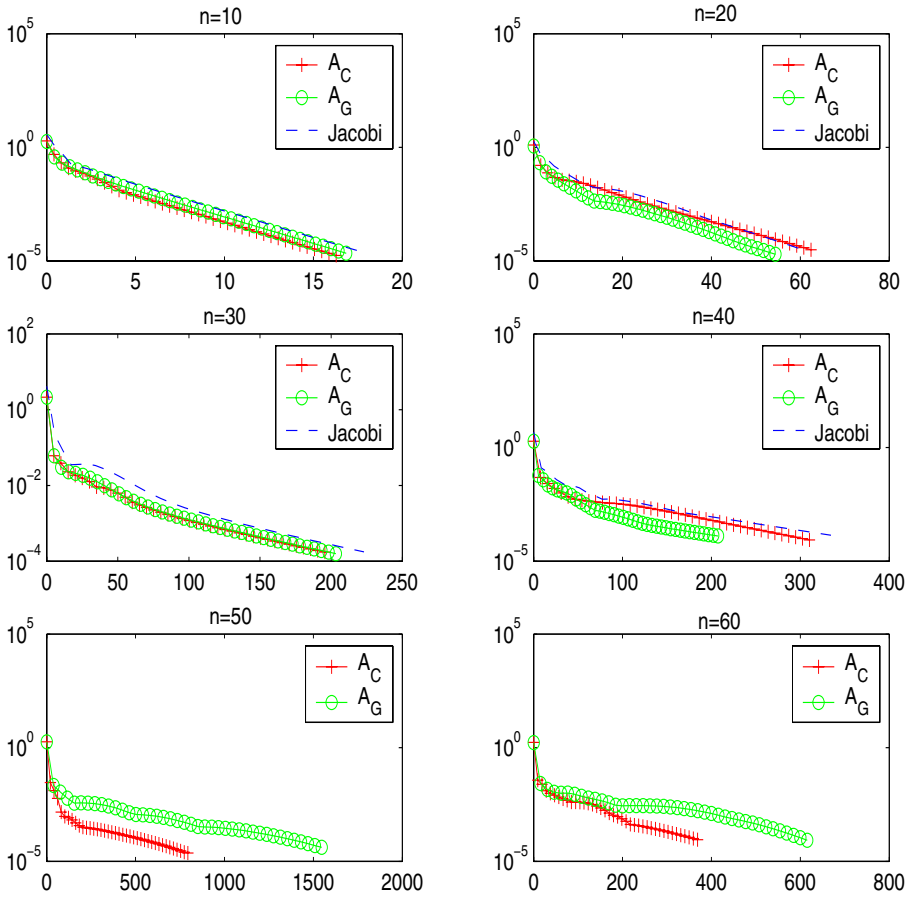


FIG. 5.2. *Convergence for* $n = 10, \ldots, 60$

Here we used a smaller stepsize $h = 0.001$ to obtain convergence. The isospectral flow (4.1) converges to the Toeplitz matrix with the first row given by

$$(185.1848,\ 181.0587,\ 169.8355,\ 154.1471,\ 136.6688,\ 119.3399).$$

In Figure 5.1 (right), the history of the convergence indicates $\mathbf{A_C}$ as the best initial value.

Example 2. In the following examples we show the convergence of the algorithm for problems of size n with eigenvalues randomly chosen.[1] We set $n = 10, 20, 30, 40$. In Figure 5.2 (first two rows), it can be seen that the isospectral flow converge monotonically in all cases, but choosing the Jacobi matrix as starting matrix is the worst strategy. We now consider larger dimensional cases by setting $n = 50, 60$. In Figure 5.2 (last row), we report the history of convergence only for starting values $\mathbf{A_C}$, $\mathbf{A_G}$ since the flow, starting with Jacobi centrosymmetric matrix, does not converge. These examples seems to suggest that the choice of $\mathbf{A_C}$ or $\mathbf{A_G}$ allows us also to manage large-dimensional problems without any additional costs.

6. Conclusions. Some inverse eigenvalue problems for Toeplitz-related structure matrices are considered in this paper. In particular, exploiting the properties of the Chebyshev polynomials, all $n!$ symmetric centrosymmetric Toeplitz-plus-Hankel matrices having (1.1) as eigenvalues are constructed. The closed form formula enables the construction of symmetric Toeplitz matrices with double eigenvalues. Furthermore, the closed formula seems to provide suitable starting values for a flow procedure that solves TIEP in the most general form. Our numerical tests confirm that the use of these starting values improve the convergence of the underlying methods.

Acknowledgment. The author would like to thank the anonymous referees for their constructive and detailed suggestions that improved significantly the paper in terms of presentations.

REFERENCES

[1] D. Bini and M. Capovani, *Spectral and computational properties of band symmetric Toeplitz matrices*, Linear Algebra Appl. 52/53 (1983), pp. 99–126.
[2] A. Cantoni and P. Butler, *Eigenvalues and eigenvectors of symmetric centrosymmetric matrices*, Linear Algebra Appl. 13 (1976), pp. 275–288.
[3] R. Chan, H. Chung, and S. Xu, *The inexact Newton-like method for inverse eigenvalue problem*, BIT, 43 (2003), pp. 7–20.
[4] M.T. Chu, *On a Differential Equation* $\frac{dX}{dt} = [X, K(X)]$ *Where K is a Toeplitz Annihilator,* preprint, North Carolina State University, Raleigh, NC, 1994.
[5] M.T. Chu, *Inverse eigenvalue problems,* SIAM Rev., 40 (1998), pp. 1–39.
[6] M.T. Chu and G.H. Golub, *Structured inverse eigenvalue problems*, Acta Numerica 11 (2002), pp. 1–71.
[7] G. Cybenko, *The numerical stability of the Levinson–Durbin algorithm for Toeplitz systems of equations,* SIAM J. Sci. Stat. Comput., 1 (1980), pp. 303–319.
[8] G. Cybenko, *On the eigenstructure of Toeplitz matrices,* IEEE Trans. Acoust., Speech, Signal Proc., ASSP, 31 (1984), pp. 910–920.
[9] G. Cybenko and C. F. Van Loan, *Computing the minimum eigenvalue of a symmetric positive definite Toeplitz matrix,* SIAM J. Sci. Stat. Comput., 7 (1986), pp 123–131.
[10] F. Diele and I. Sgura, *Centrosymmetric isospectral flows and some inverse eigenvalue problems,* Linear Algebra Appl. 366 (2003), pp 199–214.
[11] F. Diele and I. Sgura, *The Cayley method and the inverse eigenvalue problem for Toeplitz matrices,* BIT, 42 (2002), pp. 285–299.

[1] The eigenvalues values are computed by the function rand of Matlab.

[12] F. Diele and I. Sgura, *Isospectral flows and the inverse eigenvalue problem for Toeplitz matrices,* J. Comput. Appl. Math., 110 (1999), pp. 25–43.
[13] F. Diele, L. Lopez and T. Politi, *One step semi-explicit methods based on the Cayley transform for solving isospectral flows,* J. Comput. Appl. Math., 89 (1998), pp. 219–223.
[14] F. Diele, L. Lopez and R. Peluso, *The Cayley transform in the numerical solution of unitary differential systems,* Adv. Comput. Math., 8 (1998), pp. 317–334.
[15] S. Friedland, *Inverse eigenvalue problems for symmetric Toeplitz matrices,* SIAM J. Matrix Anal. Appl. 13 (1992) pp. 1142–1153.
[16] S. Friedland, J. Nocedal and M.L. Overton, *The formulation and analysis of numerical methods for inverse eigenvalue problems,* SIAM J. Numer. Anal., 24 (1987), pp. 634–667.
[17] G. Gautschi, *The condition of Vandermonde-like matrices involving orthogonal polynomials,* Linear Algebra Appl. 52/53 (1983), pp. 293–300.
[18] G.H. Golub and C. F. Van Loan, *Matrix Computations,* 3rd ed., The John Hopkins University Press, Baltimore, MD, 1996.
[19] G. Heinig, *Chebyshev–Hankel matrices and the splitting approach for centrosymmetric Toeplitz–plus–Hankel matrices,* Linear Algebra Appl., 327 (2001), pp. 181–196.
[20] Y.H. Hu and S.Y. Kung, *Computation of the minimum eigenvalue of a Toeplitz matrix by the Levinson algorithm,* Proceedings SPIE 25th International Conference (Real Time Signal Processing), San Diego, August 1980, pp. 40–45.
[21] Y.H. Hu and S.Y. Kung, *A Toeplitz eigensystem solver,* IEEE Trans. Acoust. Speech Signal Process. ASSP 33 (1986), pp. 485–491.
[22] I. C. F. Ipsen, *Computing an eigenvector with inverse iteration,* SIAM Rev., 39 (1997), pp. 254–291.
[23] H.J. Landau, *The inverse eigenvalue problem for real symmetric Toeplitz matrices,* J. Amer. Math. Soc. 7 (1994), pp. 749–767.
[24] D.P. Laurie, *A numerical approach to the inverse toeplitz eigenproblem,* SIAM J. Sci. Statist. Comput., 9 (1988), pp. 401–405.
[25] D.P. Laurie, *Solving the inverse eigenvalue problem via the eigenvector matrix,* J. Comput. Appl. Math., 35 (1991), pp. 277–289.
[26] K.R. Rao and P. Yip, *Discrete Cosine Transform. Algorithms, Advantages, Applications,* Academic Press, Boston, 1990.
[27] T.J. Rivlin, *Chebyshev Polynomials. From Approximation Theory to Algebra and Number Theory*, 2nd ed., John Wiley & Sons, New York, 1990.
[28] W.F. Trench, *Numerical solution of the eigenvalue problem for Hermitian Toeplitz matrices,* SIAM J. Matrix Anal. Appl., 10 (1989), pp. 135–146.
[29] W.F. Trench, *Numerical solution of the inverse eigenvalue problem for real symmetric Toeplitz matrices,* SIAM J. Sci. Comput., 18 (1997), pp. 1722–1736.
[30] W.F. Trench, *Interlacement of the even and odd spectra of real symmetric Toeplitz matrices,* Linear Algebra Appl., 195 (1993), pp. 59–68.
[31] C.F. Van Loan, *Computational Frameworks for the Fast Fourier Transform,* Frontiers Appl. Math. 10, SIAM, Philadelphia, 1992.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 2, pp. 295–327

COMPUTATION OF THE CANONICAL DECOMPOSITION BY MEANS OF A SIMULTANEOUS GENERALIZED SCHUR DECOMPOSITION*

LIEVEN DE LATHAUWER†, BART DE MOOR‡, AND JOOS VANDEWALLE‡

Abstract. The canonical decomposition of higher-order tensors is a key tool in multilinear algebra. First we review the state of the art. Then we show that, under certain conditions, the problem can be rephrased as the simultaneous diagonalization, by equivalence or congruence, of a set of matrices. Necessary and sufficient conditions for the uniqueness of these simultaneous matrix decompositions are derived. In a next step, the problem can be translated into a simultaneous generalized Schur decomposition, with orthogonal unknowns [A.-J. van der Veen and A. Paulraj, *IEEE Trans. Signal Process.,* 44 (1996), pp. 1136–1155]. A first-order perturbation analysis of the simultaneous generalized Schur decomposition is carried out. We discuss some computational techniques (including a new Jacobi algorithm) and illustrate their behavior by means of a number of numerical experiments.

Key words. multilinear algebra, higher-order tensor, canonical decomposition, parallel factors analysis, generalized Schur decomposition

AMS subject classifications. 15A18, 15A69

DOI. 10.1137/S089547980139786X

1. Introduction.

An increasing number of signal processing problems involves the manipulation of quantities of which the elements are addressed by more than two indices. In the literature these higher-order equivalents of vectors (first order) and matrices (second order) are called higher-order tensors, multidimensional matrices, or multiway arrays. For a lot of applications involving higher-order tensors, the existing framework of vector and matrix algebra appears to be insufficient and/or inappropriate. The algebra of higher-order tensors is called multilinear algebra.

Rank-related issues in multilinear algebra are thoroughly different from their matrix counterparts. Let us first introduce some definitions. A rank-1 tensor is a tensor that consists of the outer product of a number of vectors. For an Nth-order tensor $\mathcal{A}$ and N vectors $U^{(1)}$, $U^{(2)}$, ..., $U^{(N)}$, this means that $a_{i_1 i_2 \dots i_N} = u^{(1)}_{i_1} u^{(2)}_{i_2} \dots u^{(N)}_{i_N}$ for all values of the indices, which will be concisely written as $\mathcal{A} = U^{(1)} \circ U^{(2)} \circ \dots \circ U^{(N)}$. An n-mode vector of an $(I_1 \times I_2 \times \dots \times I_N)$-tensor $\mathcal{A}$ is an I_n-dimensional vector obtained from $\mathcal{A}$ by varying the index i_n and keeping the other indices fixed. The n-rank of a higher-order tensor is the obvious generalization of the column (row) rank of matrices: it equals the dimension of the vector space spanned by the n-mode vec-

*Received by the editors November 12, 2001; accepted for publication (in revised form) by D. P. O'Leary, November 21, 2003; published electronically November 17, 2004. This research was supported by (1) the Flemish Government: (a) Research Council K.U.Leuven: GOA-MEFISTO-666, IDO, (b) the Fund for Scientific Research-Flanders (F.W.O.) projects G.0240.99, G.0115.01, G.0197.02, and G.0407.02, (c) the F.W.O. Research Communities ICCoS and ANMMM, (d) project BIL 98 with South Africa; (2) the Belgian State, Prime Minister's Office, Federal Office for Scientific, Technical and Cultural Affairs, Interuniversity Poles of Attraction Programme IUAP IV-02 and IUAP V-22.

http://www.siam.org/journals/simax/26-2/39786.html

†ETIS, UMR 8051, 6 avenue du Ponceau, BP 44, F 95014 Cergy-Pontoise Cedex, France (delathau@ensea.fr, http://www.etis.ensea.fr).

‡SCD-SISTA of the E.E. Dept. (ESAT) of the K.U.Leuven, Kasteelpark Arenberg 10, B-3001 Leuven (Heverlee), Belgium (demoor@esat.kuleuven.ac.be, vdwalle@esat.kuleuven.ac.be, http://www.esat.kuleuven.ac.be/sista-cosic-docarch/).

tors. An important difference with the rank of matrices is that the different n-ranks of a higher-order tensor are not necessarily the same. The n-rank will be denoted as $\text{rank}_n(\mathcal{A}) = R_n$. Even when all the n-ranks are the same, they can still be different from *the* rank of the tensor, denoted as $\text{rank}(\mathcal{A}) = R$; $\mathcal{A}$ having rank R generally means that it can be decomposed in a sum of R, but not less than R, rank-1 terms; see, e.g., [34].

Example 1. Consider the $(2 \times 2 \times 2)$-tensor $\mathcal{A}$ defined by

$$\begin{cases} a_{111} = a_{112} = 1, \\ a_{221} = a_{222} = 2, \\ a_{211} = a_{121} = a_{212} = a_{122} = 0. \end{cases}$$

The 1-mode vectors are the columns of the matrix

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 2 \end{pmatrix}.$$

Because of the symmetry, the set of 2-mode vectors is the same as the set of 1-mode vectors. The 3-mode vectors are the columns of the matrix

$$\begin{pmatrix} 1 & 0 & 0 & 2 \\ 1 & 0 & 0 & 2 \end{pmatrix}.$$

Hence, we have that $R_1 = R_2 = 2$ but $R_3 = 1$.

Example 2. Consider the $(2 \times 2 \times 2)$-tensor $\mathcal{A}$ defined by

$$\begin{cases} a_{211} = a_{121} = a_{112} = 1, \\ a_{111} = a_{222} = a_{122} = a_{212} = a_{221} = 0. \end{cases}$$

The 1-rank, 2-rank, and 3-rank are equal to 2. The rank, however, equals 3, since

$$\mathcal{A} = E_2 \circ E_1 \circ E_1 + E_1 \circ E_2 \circ E_1 + E_1 \circ E_1 \circ E_2,$$

in which

$$E_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad E_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

is a decomposition in a minimal linear combination of rank-1 tensors (a proof is given in [17]).

The scalar product $\langle \mathcal{A}, \mathcal{B} \rangle$ of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is defined in a straightforward way as $\langle \mathcal{A}, \mathcal{B} \rangle \overset{\text{def}}{=} \sum_{i_1} \sum_{i_2} \ldots \sum_{i_N} a_{i_1 i_2 \ldots i_N} b_{i_1 i_2 \ldots i_N}$. The Frobenius-norm of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is then defined as $\|\mathcal{A}\| \overset{\text{def}}{=} \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$. Two tensors are called orthogonal when their scalar product is zero.

In [19] we discussed a possible multilinear generalization of the singular value decomposition (SVD). The different n-rank values can easily be read from this decomposition. In [20] we examined some techniques to compute the least-squares approximation of a given tensor by a tensor with prespecified n-ranks. On the other hand, in [19] we emphasized that the decomposition that was being studied, is not necessarily rank-revealing. This is a drawback of unitary (orthogonal) tensor decompositions in general. In this paper we will study the decomposition of a given tensor as a linear combination of a minimal number of possibly nonorthogonal, rank-1 terms. This type

of decomposition is often called "canonical decomposition" (CANDECOMP) or "parallel factors" model (PARAFAC). It is a multilinear generalization of diagonalizing a matrix by an equivalence or congruence transformation. However, it has thoroughly different properties, e.g., as far as uniqueness is concerned.

Section 2 is a brief introduction to the subject, with a formal definition of the CANDECOMP-concept and an overview of the main current computational techniques. In this section we will also mark out the problem that we will consider in this paper (we will make some specific assumptions concerning the linear independence of the canonical components). In section 3 we discuss a preprocessing step that allows us to reduce the dimensionality of the problem. In section 4 we establish a computational link between the tensor decomposition and the simultaneous diagonalization of a set of matrices by equivalence or congruence; this problem might also be looked at as a simultaneous matrix eigenvalue decomposition (EVD). The fact that the CANDECOMP usually involves nonorthogonal factor matrices is numerically disadvantageous. By reformulating the problem as a simultaneous generalized Schur decomposition (SGSD), the unknowns are restricted to the manifold of orthogonal matrices in section 5. In section 6 we discuss the advantage of working via a simultaneous matrix decomposition as opposed to working via a single EVD; this section also contains a first-order perturbation analysis of the SGSD. Techniques for the actual computation of the SGSD are considered in section 7. In section 8 it is explained how the original CANDECOMP-components can be retrieved from the components of the SGSD. In section 9 the different techniques are illustrated by means of a number of numerical experiments.

This paper contains the following new contributions:

- In the literature one finds that, in theory, the CANDECOMP can be computed by means of a matrix EVD (under the uniqueness assumptions specified in section 2) [38, 43, 5, 42]. We show that one can actually interpret the tensor decomposition as a *simultaneous* matrix decomposition. The simultaneous matrix decomposition is numerically more robust than a single EVD.
- We show that the CANDECOMP can be reformulated as an *orthogonal* simultaneous matrix decomposition—the SGSD. The reformulation in terms of orthogonal unknowns allows for the application of typical numerical procedures that involve orthogonal matrices. The SGSD as such already appeared in [48]. The difference is that in this paper it is applied to unsymmetric, instead of symmetric, matrices. This generalization may raise some confusion. It might, for instance, be tempting to consider also a simultaneous lower triangularization, in addition to a simultaneous upper triangularization.
- We derive a Jacobi-algorithm for the computation of the SGSD. The formula for the determination of the rotation angle is an explicit solution for the case of rank-2 tensors.
- The way in which the canonical components are derived from the components of the SGSD is more general and more robust than the procedure proposed in [48].
- We derive necessary and sufficient conditions for the uniqueness of a number of simultaneous matrix decompositions: (1) simultaneous diagonalization by equivalence or congruence, (2) simultaneous EVD of nonsymmetric matrices, (3) simultaneous Schur decomposition (SSD).
- We conduct a first-order perturbation analysis of the SGSD.

Before starting with the next section, we add a comment on the notation that is used. To facilitate the distinction between scalars, vectors, matrices and higher-order

tensors, the type of a given quantity will be reflected by its representation: scalars are denoted by lower-case letters (a, b, ...; α, β, ...), vectors are written as capitals (A, B, ...) (italic shaped), matrices correspond to bold-face capitals ($\mathbf{A}$, $\mathbf{B}$, ...) and tensors are written as calligraphic letters ($\mathcal{A}$, $\mathcal{B}$, ...). This notation is consistently used for lower-order parts of a given structure. For instance, the entry with row index i and column index j in a matrix $\mathbf{A}$, i.e., $(\mathbf{A})_{ij}$, is symbolized by a_{ij} (also $(A)_i = a_i$ and $(\mathcal{A})_{i_1 i_2 \ldots i_N} = a_{i_1 i_2 \ldots i_N}$); furthermore, the ith column vector of a matrix $\mathbf{A}$ is denoted as A_i, i.e., $\mathbf{A} = [A_1 A_2 \ldots]$. To enhance the overall readability, we have made one exception to this rule: as we frequently use the characters i, j, r, and n in the meaning of indices (counters), I, J, R, and N will be reserved to denote the index upper bounds, unless stated otherwise.

2. The canonical decomposition. The CANDECOMP or PARAFAC model is defined as follows.

DEFINITION 2.1 (CANDECOMP). *A canonical decomposition or parallel factors decomposition of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is a decomposition of $\mathcal{A}$ as a linear combination of a minimal number of rank-1 terms*:

$$\mathcal{A} = \sum_{r}^{R} \lambda_r \, U_r^{(1)} \circ U_r^{(2)} \circ \cdots \circ U_r^{(N)}. \tag{2.1}$$

The decomposition is visualized for third-order tensors in Figure 2.1.

The terminology originates from psychometrics [10] and phonetics [26]. Later on, the decomposition model was also applied in chemometrics [1]. Recently, the decomposition drew the attention of researchers in signal processing [14, 16, 45, 46]. A good tutorial of the current state of the art in psychometrics and chemometrics is [3].

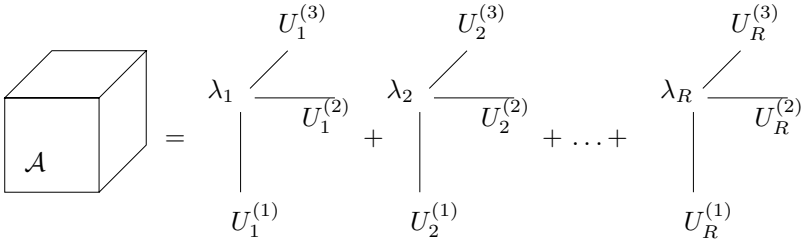


FIG. 2.1. *Visualization of the CANDECOMP for a third-order tensor.*

The decomposition can be considered as the tensorial generalization of the diagonalization of matrices by equivalence transformation (unsymmetric case) or by congruence transformation (symmetric case). However, its properties are thoroughly different from its second-order counterparts.

A first striking difference with the matrix case is that the rank of a real-valued tensor in the field of complex numbers is not necessarily equal to the rank of the same tensor in the field of real numbers [35]. Second, even if nonorthogonal rank-1 terms are allowed, the minimal number of terms is not bounded by $\min\{I_1, I_2, \ldots, I_N\}$ in general (cf. Example 2); it is usually larger and depends also on the tensor order. The determination of the maximal attainable rank value over the set of $(I_1 \times I_2 \times \cdots \times I_N)$-tensors is still an open problem in the literature. In [14] an overview of some partial results, obtained for super-symmetric tensors in the context of invariant theory, is given. (A real-valued tensor is called super-symmetric when it is invariant under arbitrary

index permutations.) The paper includes a tensor-independent rank upper-bound, an algorithm to compute maximal generic ranks and a complete discussion of the case of super-symmetric $(2 \times 2 \times \cdots \times 2)$-tensors.

The uniqueness properties of the CANDECOMP are also very different from (and much more complicated than) their matrix equivalents. The theorems of [14] allow one to determine the dimensionality of the set of valid decompositions for generic super-symmetric tensors. The deepest result concerning uniqueness of the decomposition for third-order real-valued tensors is derived from a combinatorial algebraic perspective in [34]. The complex counterpart is concisely proved in [45]. The result is generalized to arbitrary tensor orders in [47]. In [6] complex fourth-order cumulant tensors are addressed. Here we will restrict ourselves to some remarks of a more general nature, that are of direct importance to this paper. From the CANDECOMP-definition it is clear that the decomposition is insensitive to

- a permutation of the rank-1 terms,
- a scaling of the vectors $U_r^{(n)}$, combined with the inverse scaling of the coefficients λ_r.

Apart from these trivial indeterminacies, uniqueness of the CANDECOMP has been established under mild conditions of linear independence (see further for a precise formulation of the conditions imposed in this paper). Contrarily, the decomposition of a matrix $\mathbf{A}$ in a sum of rank($\mathbf{A}$) rank-1 terms is usually made unique by imposing stronger (e.g., orthogonality) constraints. In addition, for an essentially unique CANDECOMP the number of terms R can exceed $\min\{I_1, I_2, \ldots, I_N\}$.

Example 3. Consider the $(2 \times 2 \times 2)$-tensor $\mathcal{A}$ defined by

$$\begin{cases} a_{111} = a_{121} = -a_{212} = -a_{222} = 3, \\ a_{221} = a_{112} = -a_{211} = -a_{122} = 1. \end{cases}$$

The CANDECOMP of this tensor is given by

$$\mathcal{A} = X_1 \circ Y_1 \circ Z_1 + X_2 \circ Y_2 \circ Z_2, \tag{2.2}$$

in which

$$X_1 = Z_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad Z_1 = X_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad Y_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad Y_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

Apart from the trivial indeterminacies described above, this decomposition is unique, as will become clear in section 4. The reason is that the matrices $\mathbf{X} = (X_1\, X_2)$, $\mathbf{Y} = (Y_1\, Y_2)$, and $\mathbf{Z} = (Z_1\, Z_2)$ are each nonsingular.

On the other hand, consider the first "matrix slice" of $\mathcal{A}$ (cf. Figure 4.1):

$$\mathbf{A}_1 = \begin{pmatrix} a_{111} & a_{121} \\ a_{211} & a_{221} \end{pmatrix} = \begin{pmatrix} 3 & 3 \\ -1 & 1 \end{pmatrix}.$$

Due to (2.2), we have that

$$\mathbf{A}_1 = X_1 Y_1^T + X_2 Y_2^T = \mathbf{X} \cdot \mathbf{Y}^T,$$

but this decomposition is not unique. As a matter of fact, one can write

$$\mathbf{A}_1 = (\mathbf{X}\mathbf{F}) \cdot (\mathbf{Y}\mathbf{F}^{-T})^T = \tilde{\mathbf{X}} \cdot \tilde{\mathbf{Y}}^T,$$

for any nonsingular (2×2) matrix $\mathbf{F}$. One way to make this decomposition essentially unique, is to claim that the columns of $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ are orthogonal. The solution is then given by the SVD of $\mathbf{A}_1$.

It is a common practice to look for the CANDECOMP-components by straightforward minimization of the quadratic cost function

$$f(\hat{\mathcal{A}}) = \|\mathcal{A} - \hat{\mathcal{A}}\|^2 \tag{2.3}$$

over all rank-R tensors $\hat{\mathcal{A}}$, which we will parametrize as

$$\hat{\mathcal{A}} = \sum_{r}^{R} \hat{\lambda}_r \, \hat{U}_r^{(1)} \circ \hat{U}_r^{(2)} \circ \cdots \circ \hat{U}_r^{(N)}. \tag{2.4}$$

It is possible to resort to an alternating least-squares (ALS) algorithm, in which the vector estimates are updated mode per mode [10]. The idea is as follows. Let us define

$$\hat{\mathbf{U}}^{(n)} \stackrel{\text{def}}{=} [\hat{U}_1^{(n)} \hat{U}_2^{(n)} \ldots \hat{U}_R^{(n)}],$$
$$\hat{\mathbf{\Lambda}} \stackrel{\text{def}}{=} \text{diag}\{(\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_R)\},$$

in which diag$\{\cdot\}$ is a diagonal matrix, containing the entries of its argument on the diagonal. If we now imagine that the matrices $\hat{\mathbf{U}}^{(m)}$, $m \neq n$, are fixed, then (2.3) is merely a quadratic expression in the components of the matrix $\hat{\mathbf{U}}^{(n)} \cdot \hat{\mathbf{\Lambda}}$; the estimation of these components is a classical linear least-squares problem. An ALS iteration consists of repeating this procedure for different mode numbers: in each step the estimate of one of the matrices $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}$ is optimized, while the other matrix estimates are kept constant. Overflow and underflow can be avoided by normalizing the estimates of the columns $U_r^{(n)}$ $(1 \leqslant r \leqslant R; 1 \leqslant n \leqslant N)$ to unit-length.

For $R = 1$, the ALS algorithm can be interpreted as a generalization of the power method for the computation of the best rank-1 approximation of a matrix [20]. For $R > 1$, however, the canonical components can in principle not be obtained by means of a deflation algorithm. The reason is that the stationary points of the higher-order power iteration generally do not correspond to one of the terms in (2.4), and that the residue is in general not of rank $R - 1$ [32]. This even holds when the rank-1 terms are mutually orthogonal [33]. Only when each of the matrices $\{\mathbf{U}^{(n)}\}$ is column-wise orthonormal, the deflation approach will work, but in this special case, the components can be obtained by means of a matrix SVD [19].

Because the cost function is monotonically decreasing, one expects that the ALS algorithm converges to a (local) minimum of $f(\hat{\mathcal{A}})$. If the CANDECOMP-model is only approximately valid, the risk of finding a spurious local optimum can be diminished by repeating the optimization for a number of randomly chosen initial values. The decision on whether the global optimum has been found or not usually relies on heuristics. The process of iterating over different starting values can be time-consuming. In addition, if the directions of some of the n-mode vectors in the CANDECOMP-model $(1 \leqslant n \leqslant N)$ are close, then it seems unlikely that this configuration is found from a random start [14]. Some alternative initializations are discussed in [11]. The rank itself is usually determined by repeating the procedure for different values of R, and comparing the results. An alternative, also based on heuristics, is the evaluation of split-half experiments [27].

ALS iterations can be very slow. In addition, it is sometimes observed that the algorithm moves through a "swamp": the algorithm seems to converge, but then the convergence speed drastically decreases and remains small for several iteration steps, after which it may suddenly increase again. The nature of swamps and how they can be avoided forms a topic of ongoing research [41, 36]. To cope with the slow convergence, a number of acceleration methods have been proposed [26, 28, 31]. One could make use of a prediction technique, in which estimates of previous iteration steps are extrapolated to forecast new estimates [3].

In [40] a Gauss–Newton method is described, in which all the CANDECOMP-factors are updated simultaneously; in addition, the inherent indeterminacy of the decomposition has been fixed by adding a quadratic regularization constraint on the component entries.

On the other hand, setting the gradient of f to zero and solving the resulting set of equations, is computationally hard as well: a set of $R(I_1 + I_2 + \ldots I_N) - R(N-1)$ polynomial equations of degree $2N-1$, in $R(I_1 + I_2 + \ldots I_N) - R(N-1)$ independent unknowns, has to be solved (to determine this dimensionality, imagine that the indeterminacy has been overcome by incorporating the factor λ_r $(1 \leqslant r \leqslant R)$ in one of the vectors of the rth outer product, and by fixing one nonzero entry in the other vectors).

An interesting alternative procedure, which works under a number of assumptions among which the most restrictive is that $R \leqslant \min\{I_1, I_2\}$, has been proposed in [38]. Similar results have been proposed in [43, 5, 42]. It was explained that, if (2.1) is exactly valid, the decomposition can be found by a simple matrix EVD. When $\mathcal{A}$ is only known with limited accuracy, a least-squares matching of both sides of (2.1) can now be initialized with the EVD result. This technique forms the starting point for the developments in section 4.

Some promising computation schemes, at this moment only formulated in terms of (super-symmetric) cumulant tensors, have been developed as means to solve the problem of higher-order-only independent component analysis. In [7] Cardoso shows that under mild conditions the matrices in the intersection of the range of the cumulant tensor and the manifold of rank-1 matrices take the form of an outer product of a steering vector with itself; consequently MUSIC-like [44] algorithms are devised. In [6] the same author investigates the link between symmetry of the cumulant tensor and the rank-1 property of its components. The problem is subsequently reformulated in terms of a matrix EVD.

The decomposition of a dataset as a sum of rank-1 terms is sometimes called the *factor analysis* problem. With the decomposition, one aims at relating the different rank-1 terms to the different "physical mechanisms" that have contributed to the dataset. We repeat that factor analysis of matrices is, as such, essentially underdetermined. The extra conditions (maximal variance, orthonormality, etc.) that are usually imposed to guarantee uniqueness, are often physically irrelevant. In a wide range of parameters, this is not the case for the higher-order decomposition; the weaker conditions of linear independence to ensure uniqueness often have a physical meaning. This makes the CANDECOMP of higher-order tensors to an important signal processing tool.

In this paper, we will study the special but important case of an $(I_1 \times I_2 \times I_3)$-tensor $\mathcal{A}$ with rank $R \leqslant \min\{I_1, I_2\}$ and 3-rank $R_3 \geqslant 2$. (If $R_3 = 1$, then the different matrices obtained from $\mathcal{A}$ by fixing the index i_3 are proportional, and the CANDECOMP reduces to the diagonalization of one of these matrices by congruence

or equivalence.) We assume that

(i) the set $\{U_r^{(1)}\}_{(1\leqslant r\leqslant R)}$ is linearly independent (i.e., no vector can be written as a linear combination of the other vectors),

(ii) the set $\{U_r^{(2)}\}_{(1\leqslant r\leqslant R)}$ is linearly independent,

(iii) the set $\{U_r^{(3)}\}_{(1\leqslant r\leqslant R)}$ does not contain collinear vectors (i.e., no vector is a scalar multiple of an other vector).

Roughly speaking, we address the case in which the number of rank-1 terms is bounded by the second largest dimension of $\mathcal{A}$ (like in classical matrix decompositions). Conditions (i)–(iii) are generically satisfied, i.e., only in a set of Lebesgue measure zero they do not hold. In typical applications one has the prior knowledge that these assumptions are valid. Classical (not overcomplete) independent component analysis can be formulated in terms of this model [13, 49]. Conditions (i)–(iii) are required for the uniqueness of the solution. All the examples in the tutorial [3] belong to our class of interest. In chemometrical applications such as the ones described in [42], the conditions do not pose any problem. For instance, I_1 and I_2 correspond to the length of emission-excitation spectra and R is the number of chemical components.

If the rank of $\mathcal{A}$ is higher than $R_{\max} = \min\{I_1, I_2\}$, then our method will still try to fit a rank-$R_{\max}$ model to the data. Contrary to the matrix case, this does not simply correspond to discarding the rank-1 terms that have the smallest norm.

It can be verified that conditions (i)–(iii) are sufficient to make the CANDECOMP essentially unique [38] (see also sections 4 and 6). The exposition is restricted to real-valued third-order tensors for notational convenience. The generalization to higher tensor orders is straightforward. The method then applies to tensors of which the rank $R \leqslant \min\{I_1, I_2\}$ and at least one of the n-ranks R_n, for $n \geqslant 3$, satisfies $R_n \geqslant 2$. Conditions (i)–(iii) should be rephrased as the following:

(i) the set $\{U_r^{(1)}\}_{(1\leqslant r\leqslant R)}$ is linearly independent,

(ii) the set $\{U_r^{(2)}\}_{(1\leqslant r\leqslant R)}$ is linearly independent,

(iii) and at least one of the sets $\{U_r^{(n)}\}_{(1\leqslant r\leqslant R)}$ for $n \geqslant 3$ does not contain collinear vectors.

Apart from section 7.2, the generalization to complex-valued tensors is also straightforward. An outline of the exposition is presented as Algorithm 1. In this algorithm we assume that a value of R is given or that the rank has been estimated as $\text{rank}_1(\mathcal{A}) = \text{rank}_2(\mathcal{A})$ (see next section).

3. Dimensionality reduction. Under the assumptions specified in the previous section, we have that $R_1 = \text{rank}_1(\mathcal{A}) = R = \text{rank}_2(\mathcal{A}) = R_2$ and that $R_3 = \text{rank}_3(\mathcal{A}) = \text{rank}(\mathbf{U}^{(3)})$. To understand this, remark that (2.1) implies that the n-mode vectors of $\mathcal{A}$ are the columns of the matrix

$$\mathbf{A}_{(n)} = \mathbf{U}^{(n)} \cdot \mathbf{\Lambda} \cdot (\mathbf{U}^{(m)} \odot \mathbf{U}^{(l)})^T,$$

in which $\odot$ is the Kathri–Rao or columnwise Kronecker product and (n, m, l) is an arbitrary permutation of $(1, 2, 3)$. Hence, conditions (i)–(iii) imply that the dimension of the n-mode vector space, which equals the rank of $\mathbf{A}_{(n)}$, is equal to $\text{rank}(\mathbf{U}^{(n)})$.

If $R < \max\{I_1, I_2\}$, or $R_3 < I_3$, then an a priori dimensionality reduction of $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ to a tensor $\mathcal{B} \in \mathbb{R}^{R \times R \times R_3}$ decreases the computational load of the actual determination of the CANDECOMP (step 1 in Algorithm 1). Before starting the actual exposition, we briefly address this issue. Suppose that $\mathcal{A}$ and $\mathcal{B}$ are related

ALGORITHM 1
CANDECOMP BY SGSD
In: $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, R.
Out: $\{U_r^{(1)}\}_{(1 \leqslant r \leqslant R)}$, $\{U_r^{(2)}\}_{(1 \leqslant r \leqslant R)}$, $\{U_r^{(3)}\}_{(1 \leqslant r \leqslant R)}$, $\{\lambda_r\}_{(1 \leqslant r \leqslant R)}$ such that $\mathcal{A} \simeq \sum_r^R \lambda_r U_r^{(1)} \circ U_r^{(2)} \circ U_r^{(3)}$.

1. Perform an initial best rank-(R, R, R_3) approximation of $\mathcal{A}$: maximize $g(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)}) = \|\mathcal{A} \times_1 \mathbf{X}^{(1)^T} \times_2 \mathbf{X}^{(2)^T} \times_3 \mathbf{X}^{(3)^T}\|^2$ over columnwise orthonormal $\mathbf{X}^{(1)} \in \mathbb{R}^{I_1 \times R}$, $\mathbf{X}^{(2)} \in \mathbb{R}^{I_2 \times R}$ and $\mathbf{X}^{(3)} \in \mathbb{R}^{I_3 \times R_3}$; $\max(g(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)})) = g(\mathbf{X}^{(1)}_{\max}, \mathbf{X}^{(2)}_{\max}, \mathbf{X}^{(3)}_{\max})$. $\mathcal{B} = \mathcal{A} \times_1 \mathbf{X}^{(1)^T}_{\max} \times_2 \mathbf{X}^{(2)^T}_{\max} \times_3 \mathbf{X}^{(3)^T}_{\max}$. Continue for $\mathcal{B}$ with steps 2, 3, 4, (5) below. $\hat{\mathcal{A}} = \hat{\mathcal{B}} \times_1 \mathbf{X}^{(1)}_{\max} \times_2 \mathbf{X}^{(2)}_{\max} \times_3 \mathbf{X}^{(3)}_{\max}$. (section 3.) (Perform step 5 for $\hat{\mathcal{A}}$.))
2. Associate to $\mathcal{A}$ a linear mapping $f_{\mathcal{A}}$ from $\mathbb{R}^{I_3}$ to $\mathbb{R}^{I_1 \times I_2}$ (see (4.1)). Determine $\{\mathbf{V}_k\}_{(1 \leqslant k \leqslant K)}$ such that the range of $f_{\mathcal{A}}$ is spanned by $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_K$.
3. Compute orthogonal $\mathbf{Q}, \mathbf{Z}$ and (approximately) upper triangular $\{\mathbf{R}_k\}_{(1 \leqslant k \leqslant K)}$ from the SGSD of (5.1)–(5.3):
   - extended QZ-iteration (section 7.1, [48]), or
   - Jacobi-type iteration (section 7.2, [17, 18]).
4. Compute $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ from $\{\mathbf{R}_k\}_{(1 \leqslant k \leqslant K)}$ and $\{\mathbf{V}_k\}_{(1 \leqslant k \leqslant K)}$ (and $\mathbf{Q}$, $\mathbf{Z}$). Compute $\mathbf{U}^{(3)}$ from $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$ and $\mathcal{A}$. (Detailed outline in section 8.)
5. Minimize $f(\hat{\mathcal{A}}) = \|\mathcal{A} - \hat{\mathcal{A}}\|^2$ (section 2).)

by

$$a_{i_1 i_2 i_3} = \sum_{r_1 r_2 r_3} x^{(1)}_{i_1 r_1} x^{(2)}_{i_2 r_2} x^{(3)}_{i_3 r_3} b_{r_1 r_2 r_3} \tag{3.1}$$

for all index values, where $\mathbf{X}^{(1)} \in \mathbb{R}^{I_1 \times R}$, $\mathbf{X}^{(2)} \in \mathbb{R}^{I_2 \times R}$ and $\mathbf{X}^{(3)} \in \mathbb{R}^{I_3 \times R_3}$, which we will write concisely as

$$\mathcal{A} = \mathcal{B} \times_1 \mathbf{X}^{(1)} \times_2 \mathbf{X}^{(2)} \times_3 \mathbf{X}^{(3)}. \tag{3.2}$$

If $\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$, $\mathbf{X}^{(3)}$ each have mutually orthonormal columns, then the optimal rank-R approximation $\hat{\mathcal{B}}$ of $\mathcal{B}$ and the optimal rank-R approximation $\hat{\mathcal{A}}$ of $\mathcal{A}$ are related in the same way:

$$\hat{\mathcal{A}} = \hat{\mathcal{B}} \times_1 \mathbf{X}^{(1)} \times_2 \mathbf{X}^{(2)} \times_3 \mathbf{X}^{(3)}, \tag{3.3}$$

since "n-mode multiplication" with the columnwise orthonormal matrices $\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$, $\mathbf{X}^{(3)}$ does not change the cost function f (2.3). If the CANDECOMP-model is exactly satisfied, then any orthonormal basis of the mode-1, mode-2, and mode-3 vectors of $\mathcal{A}$ gives a suitable $\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$, $\mathbf{X}^{(3)}$, respectively. In practice, however, $R = R_1 = R_2$ and R_3 will be estimated as the number of significant mode-1 / mode-2 and mode-3 singular values of $\mathcal{A}$ (see [19]). An optimal rank-(R, R, R_3) approximation of $\mathcal{A}$, before computing the optimal rank-R approximation, can then be realized. For techniques we refer to [20].

4. CANDECOMP and simultaneous EVD. Without loss of generality we assume that $I_1 = I_2 = R$ (if $I_1 > R$ or $I_2 > R$, we can always do a dimensionality

reduction, as explained in the previous section). We start the derivation of our computation scheme with associating to $\mathcal{A}$ a linear transformation of the vector space $\mathbb{R}^{I_3}$ to the matrix space $\mathbb{R}^{I_1 \times I_2}$, in the following way:

$$\mathbf{V} = f_{\mathcal{A}}(W) = \mathcal{A} \times_3 W \quad \Longleftrightarrow \quad v_{i_1 i_2} = \sum_{i_3} a_{i_1 i_2 i_3} w_{i_3}, \tag{4.1}$$

for all index values. Substitution of (4.1) in (2.1) shows that the image of W can easily be expressed in terms of the CANDECOMP-components:

$$\mathbf{V} = \mathbf{U}^{(1)} \cdot \mathbf{D} \cdot \mathbf{U}^{(2)^T}, \tag{4.2}$$

in which we have used the following notations:

$$\mathbf{U}^{(n)} \stackrel{\text{def}}{=} [U_1^{(n)} U_2^{(n)} \dots U_{I_n}^{(n)}], \tag{4.3}$$

$$\mathbf{D} \stackrel{\text{def}}{=} \text{diag}\{(\lambda_1, \lambda_2, \dots, \lambda_R)\} \cdot \text{diag}\{\mathbf{U}^{(3)^T} W\}. \tag{4.4}$$

Any matrix in the range of the mapping $f_{\mathcal{A}}$ can be diagonalized by equivalence with the matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$. (If $\mathcal{A}$ does not change under permutation of its first two indices, then any matrix in the range can be diagonalized by congruence with the matrix $\mathbf{U}^{(1)} = \mathbf{U}^{(2)}$.) If the range is spanned by the matrices $\mathbf{V}_1$, $\mathbf{V}_2$, ..., $\mathbf{V}_K$, then we should solve the following simultaneous decomposition:

$$\mathbf{V}_1 = \mathbf{U}^{(1)} \cdot \mathbf{D}_1 \cdot \mathbf{U}^{(2)^T}, \tag{4.5}$$

$$\mathbf{V}_2 = \mathbf{U}^{(1)} \cdot \mathbf{D}_2 \cdot \mathbf{U}^{(2)^T}, \tag{4.6}$$

$$\vdots$$

$$\mathbf{V}_K = \mathbf{U}^{(1)} \cdot \mathbf{D}_K \cdot \mathbf{U}^{(2)^T}, \tag{4.7}$$

in which $\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_K$ are diagonal. A possible choice of $\{\mathbf{V}_k\}_{(1 \leqslant k \leqslant K)}$ consists of the "matrix slices" $\{\mathbf{A}_i\}_{(1 \leqslant i \leqslant I_3)}$, obtained by fixing the index i_3 to i (see Figure 4.1); the corresponding vectors $\{W_i\}_{(1 \leqslant i \leqslant I_3)}$ are the canonical unit vectors. An other possible choice consists of the K dominant left singular matrices of the mapping in (4.1). In both cases, the cost function

$$\tilde{f}(\hat{\mathbf{U}}^{(1)}, \hat{\mathbf{U}}^{(2)}, \{\hat{\mathbf{D}}_k\}) = \sum_k \| \mathbf{V}_k - \hat{\mathbf{U}}^{(1)} \cdot \hat{\mathbf{D}}_k \cdot \hat{\mathbf{U}}^{(2)^T} \|^2$$

corresponds to the CANDECOMP cost function (2.3). The latter choice follows naturally from the analysis in section 3 [20].

For later use, we define

$$\tilde{\mathbf{U}}^{(3)} = \begin{pmatrix} (\mathbf{D}_1)_{11} & (\mathbf{D}_1)_{22} & \dots & (\mathbf{D}_1)_{RR} \\ (\mathbf{D}_2)_{11} & (\mathbf{D}_2)_{22} & \dots & (\mathbf{D}_2)_{RR} \\ \vdots & \vdots & & \vdots \\ (\mathbf{D}_K)_{11} & (\mathbf{D}_K)_{22} & \dots & (\mathbf{D}_K)_{RR} \end{pmatrix} \tag{4.8}$$

$$= [W_1 W_2 \dots W_K]^T \cdot \mathbf{U}^{(3)} \cdot \text{diag}\{(\lambda_1, \lambda_2, \dots, \lambda_R)\}. \tag{4.9}$$

If the CANDECOMP-model is exactly satisfied, then its terms can be computed from two of the equations in (4.5)–(4.7). Let us assume that the matrix $\mathbf{V}_1$ has

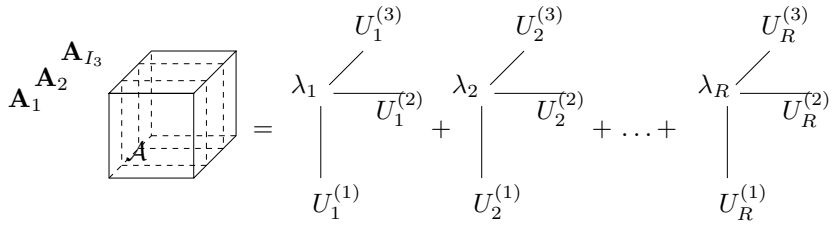


FIG. 4.1. *Definition of matrix slices for the computation of the CANDECOMP by simultaneous diagonalization.*

full rank (this is the case for a generic choice of W_1). Combination of the first two equations then leads to the following EVD:

$$\mathbf{V}_2 \cdot \mathbf{V}_1^{-1} = \mathbf{U}^{(1)} \cdot (\mathbf{D}_2 \cdot \mathbf{D}_1^{-1}) \cdot \mathbf{U}^{(1)^{-1}}. \tag{4.10}$$

Remember that we assumed in section 2 that $\mathbf{U}^{(3)}$ does not contain collinear columns. As a consequence, the pair $((\mathbf{D}_1)_{ii}(\mathbf{D}_2)_{ii}) = \lambda_i(W_1^T U_i^{(3)} W_2^T U_i^{(3)})$ and $((\mathbf{D}_1)_{jj}(\mathbf{D}_2)_{jj}) = \lambda_j(W_1^T U_j^{(3)} W_2^T U_j^{(3)})$ is generically not proportional, for all $i \neq j$. Hence the diagonal elements of $\mathbf{D}_2 \cdot \mathbf{D}_1^{-1}$ are mutually different and the EVD (4.10) reveals the columns of $\mathbf{U}^{(1)}$, up to an irrelevant scaling and/or permutation. Once $\mathbf{U}^{(1)}$ is known, $\mathbf{U}^{(2)}$ can be obtained, up to a scaling of its columns, as follows. From (4.5)–(4.7) we have

$$\mathbf{V}_1^T \cdot \mathbf{U}^{(1)^{-T}} = \mathbf{U}^{(2)} \cdot \mathbf{D}_1, \tag{4.11}$$

$$\mathbf{V}_2^T \cdot \mathbf{U}^{(1)^{-T}} = \mathbf{U}^{(2)} \cdot \mathbf{D}_2, \tag{4.12}$$

$$\vdots$$

$$\mathbf{V}_K^T \cdot \mathbf{U}^{(1)^{-T}} = \mathbf{U}^{(2)} \cdot \mathbf{D}_K. \tag{4.13}$$

Hence, if we denote the rth column of $\mathbf{V}_k^T \cdot \mathbf{U}^{(1)^{-T}}$ as B_{kr}, then $U_r^{(2)}$ can be estimated as the dominant left singular vector of $[B_{1r} B_{2r} \ldots B_{Kr}]$. Finally, the matrix $\mathbf{U}^{(3)} \cdot \mathrm{diag}\{(\lambda_1, \lambda_2, \ldots, \lambda_R)\}$ is found by solving the CANDECOMP-model as a linear set of equations, for given matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$. (Note that the assumptions that we have made for identifiability in section 2 indeed allow to obtain the CANDECOMP in an essentially unique way.) If the CANDECOMP-model is only approximately satisfied, then the estimates can be used to initialize an additional optimization algorithm for the minimization of cost function (2.3) (cf. step 5 in Algorithm 1). This EVD-approach is a variant of the techniques described in [38, 43, 5, 42].

It is intuitively clear, however, that it is preferable to exploit all the available information by taking into account all the equations in (4.5)–(4.7). This leads to a *simultaneous EVD*:

$$\mathbf{V}_2 \cdot \mathbf{V}_1^{-1} = \mathbf{U}^{(1)} \cdot (\mathbf{D}_2 \cdot \mathbf{D}_1^{-1}) \cdot \mathbf{U}^{(1)^{-1}}, \tag{4.14}$$

$$\mathbf{V}_3 \cdot \mathbf{V}_1^{-1} = \mathbf{U}^{(1)} \cdot (\mathbf{D}_3 \cdot \mathbf{D}_1^{-1}) \cdot \mathbf{U}^{(1)^{-1}}, \tag{4.15}$$

$$\vdots$$

$$\mathbf{V}_K \cdot \mathbf{V}_1^{-1} = \mathbf{U}^{(1)} \cdot (\mathbf{D}_K \cdot \mathbf{D}_1^{-1}) \cdot \mathbf{U}^{(1)^{-1}}. \tag{4.16}$$

We will further discuss the advantages in section 6.

In this paper, we propose a reliable technique to deal with (4.5)–(4.7) simultaneously (steps 2–4 in Algorithm 1).

5. CANDECOMP and SGSD. The fact that the unknown matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are basically arbitrary nonsingular matrices, makes them hard to deal with in a proper numerical way. In this section, we will reformulate the problem in terms of orthogonal unknowns. Therefore, we can make an appeal to the technique established in [48], where the symmetric equivalent of (4.5)–(4.7) was encountered in the derivation of an analytical constant modulus algorithm.

Introducing a QR-factorization $\mathbf{U}^{(1)} = \mathbf{Q}^T \mathbf{R}'$ and an RQ-decomposition $\mathbf{U}^{(2)^T} = \mathbf{R}'' \mathbf{Z}^T$ leads to a set of matrix equations that we will call a *simultaneous generalized Schur decomposition* (a set of two of the equations below is called "Generalized Schur Decomposition" [24]):

$$\mathbf{Q} \cdot \mathbf{V}_1 \cdot \mathbf{Z} = \mathbf{R}_1 = \mathbf{R}' \cdot \mathbf{D}_1 \cdot \mathbf{R}'', \tag{5.1}$$

$$\mathbf{Q} \cdot \mathbf{V}_2 \cdot \mathbf{Z} = \mathbf{R}_2 = \mathbf{R}' \cdot \mathbf{D}_2 \cdot \mathbf{R}'', \tag{5.2}$$

$$\vdots$$

$$\mathbf{Q} \cdot \mathbf{V}_K \cdot \mathbf{Z} = \mathbf{R}_K = \mathbf{R}' \cdot \mathbf{D}_K \cdot \mathbf{R}'', \tag{5.3}$$

in which $\mathbf{Q}, \mathbf{Z} \in \mathbb{R}^{R \times R}$ are orthogonal and $\mathbf{R}', \mathbf{R}'', \mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_K \in \mathbb{R}^{R \times R}$ are upper triangular. If the CANDECOMP model is exactly satisfied, the new problem consists of the determination of $\mathbf{Q}$ and $\mathbf{Z}$ such that $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_K$ are each upper triangular. In practice, this is only possible in an approximate sense. For instance, one could maximize the function g, given by

$$g(\mathbf{Q}, \mathbf{Z}) = \|\mathbf{Q} \cdot \mathbf{V}_1 \cdot \mathbf{Z}\|_{UF}^2 + \|\mathbf{Q} \cdot \mathbf{V}_2 \cdot \mathbf{Z}\|_{UF}^2 + \cdots + \|\mathbf{Q} \cdot \mathbf{V}_K \cdot \mathbf{Z}\|_{UF}^2, \tag{5.4}$$

in which $\|\cdot\|_{UF}$ denotes the Frobenius-norm of the upper triangular part of a matrix. So we will determine $\mathbf{Q}$ and $\mathbf{Z}$ as the orthogonal matrices that make $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_K$ simultaneously as upper triangular as possible. Equivalently, one may minimize

$$h(\mathbf{Q}, \mathbf{Z}) = \|\mathbf{Q} \cdot \mathbf{V}_1 \cdot \mathbf{Z}\|_{LFs}^2 + \|\mathbf{Q} \cdot \mathbf{V}_2 \cdot \mathbf{Z}\|_{LFs}^2 + \cdots + \|\mathbf{Q} \cdot \mathbf{V}_K \cdot \mathbf{Z}\|_{LFs}^2 \tag{5.5}$$

$$= \sum_k \|\mathbf{V}_k\|^2 - g(\mathbf{Q}, \mathbf{Z}), \tag{5.6}$$

in which $\|\cdot\|_{LFs}$ denotes the Frobenius-norm of the strictly lower triangular part of a matrix. The decomposition is visualized in Figure 5.1.

In section 7 we will discuss two algorithms for the computation of the SGSD. In section 8 we will explain how $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ can be calculated once $\mathbf{Q}$ and $\mathbf{Z}$ have been estimated.

Remark 4. At first sight the unsymmetric case allows for the derivation of an additional set of equations if we substitute a QL-factorization $\mathbf{U}^{(1)} = \tilde{\mathbf{Q}}^T \mathbf{L}'$ and an LQ-decomposition $\mathbf{U}^{(2)^T} = \mathbf{L}'' \tilde{\mathbf{Z}}^T$ in (4.5)–(4.7) ($\mathbf{L}'$ and $\mathbf{L}''$ are lower triangular). This leads to a simultaneous lower triangularization of the matrices $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_K$. Both approaches are in fact equivalent because they simply correspond to a different permutation of the columns of $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$, which cannot be determined in advance. Since the aim of the algorithms that will be discussed in section 7 is only to find matrices $\mathbf{Q}$ and $\mathbf{Z}$ that correspond to an arbitrary column permutation (not necessarily the one that happens to globally minimize the cost function h in the presence of noise), both formulations may in practice lead to results that are close but not exactly equal.

Remark 5. In [49] an alternative scheme, in which one directly works with the components of (4.5)–(4.7), instead of going via a SGSD, was formulated for the symmetric case, i.e., $\mathbf{U}^{(1)} = \mathbf{U}^{(2)} = \mathbf{U}$. Before continuing with the actual exposition, let

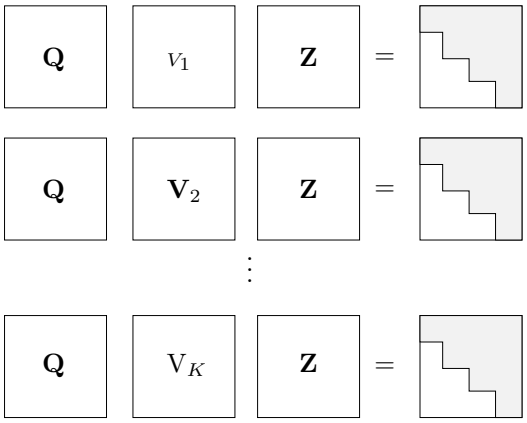


FIG. 5.1. *Visualization of a SGSD.*

us briefly address this approach. It is an ALS strategy, with the particular problem that for two of the modes the components are equal. The technique is called the "AC–DC" algorithm, standing for "alternating columns–diagonal centers". Let us associate with (4.5)–(4.7) the following weighted cost function:

$$c(\mathbf{U}, \mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_K) = \sum_{k=1}^{K} w_k \|\mathbf{V}_k - \mathbf{U} \cdot \mathbf{D}_k \cdot \mathbf{U}^T\|^2. \tag{5.7}$$

Note that for $w_k = 1$ $(1 \leqslant k \leqslant K)$ and $\{\mathbf{V}_k\}_{(1 \leqslant k \leqslant K)}$ equal to the matrix slices $\{\mathbf{A}_i\}_{(1 \leqslant i \leqslant I_3)}$ defined in Figure 4.1, this cost function corresponds to the obvious CANDECOMP cost (2.3). In the technique one alternates between updates of $\{\mathbf{D}_k\}_{(1 \leqslant k \leqslant K)}$, given $\mathbf{U}$ (DC-phase) and updates of $\mathbf{U}$, given $\{\mathbf{D}_k\}_{(1 \leqslant k \leqslant K)}$ (AC-phase). It is clear that a DC-step amounts to a linear least-squares problem. In [49] it is shown that the conditional update of a column of $\mathbf{U}$ amounts to the best rank-1 approximation of a symmetric $(I \times I)$-matrix $(I = I_1 = I_2)$. An AC-phase then consists of one, or more, updates of the different columns of $\mathbf{U}$.

6. Single vs. simultaneous decomposition and perturbation analysis. Before introducing some algorithms for the computation of the SGSD, we will discuss in this section some advantages of the simultaneous decomposition approach over the computation of a single EVD (cf. [38, 43, 5, 42]). In this context, we will also provide a first-order perturbation analysis of the SGSD.

6.1. Uniqueness. First, let us reconsider (4.14)–(4.16). One could solve these EVDs separately, and retain the solution that leads to the best CANDECOMP-estimate. However, it is safer from a numerical point of view to solve (4.14)–(4.16) simultaneously, in some optimal sense, especially when the perturbation of the matrices $\{\mathbf{V}_k\}_{(1 \leqslant k \leqslant K)}$ (with respect to their ideal values in an exact CANDECOMP) may have caused eigenvalues to cross each other. This is illustrated in the next example; a symmetric version of the example can be found in [4].

Example 6. Consider the following matrix pair:

$$\mathbf{M}_1 = \begin{pmatrix} 1-\epsilon & 0 & 0 & 0 \\ 0 & 1+\epsilon & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 3 \end{pmatrix}, \quad \mathbf{M}_2 = \begin{pmatrix} 2 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1-\epsilon & 0 \\ 0 & 0 & 0 & 1+\epsilon \end{pmatrix},$$

in which $\epsilon \in \mathbb{R}$ is small. For $\epsilon = 0$, the two matrices have a common eigenmatrix:

$$\mathbf{E} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

If $\epsilon \neq 0$, $\mathbf{E}$ still nearly diagonalizes $\mathbf{V}_1$ and $\mathbf{V}_2$:

$$\mathbf{M}_1 \cdot \mathbf{E} = \mathbf{E} \cdot \operatorname{diag}\{[1\ 1\ 2\ 3]\} + O(\epsilon), \qquad \mathbf{M}_2 \cdot \mathbf{E} = \mathbf{E} \cdot \operatorname{diag}\{[2\ 3\ 1\ 1]\} + O(\epsilon).$$

On the other hand, for $\epsilon \neq 0$, the distinct eigenmatrices $\mathbf{E}_1$ and $\mathbf{E}_2$, of $\mathbf{V}_1$ and $\mathbf{V}_2$, respectively, are not suitable for diagonalization of the other matrix:

$$\mathbf{M}_1 \cdot \mathbf{E}_2 = \mathbf{E}_2 \cdot \operatorname{diag}\{[1\ 1\ 2\ 3]\} + O(1), \qquad \mathbf{M}_2 \cdot \mathbf{E}_1 = \mathbf{E}_1 \cdot \operatorname{diag}\{[2\ 3\ 1\ 1]\} + O(1).$$

For a simultaneous EVD we have the following uniqueness theorem.

THEOREM 6.1. *For given matrices* $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_L \in \mathbb{R}^{R \times R}$*, the simultaneous decomposition*

$$\mathbf{M}_1 = \mathbf{U} \cdot \mathbf{D}_1 \cdot \mathbf{U}^{-1}, \tag{6.1}$$

$$\mathbf{M}_2 = \mathbf{U} \cdot \mathbf{D}_2 \cdot \mathbf{U}^{-1}, \tag{6.2}$$

$$\vdots$$

$$\mathbf{M}_L = \mathbf{U} \cdot \mathbf{D}_L \cdot \mathbf{U}^{-1}, \tag{6.3}$$

with $\mathbf{U} \in \mathbb{R}^{R \times R}$ *nonsingular and* $\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_L \in \mathbb{R}^{R \times R}$ *diagonal, is unique up to a permutation and a scaling of the columns of* $\mathbf{U}$ *if and only if all the columns of the matrix*

$$\mathbf{D} = \begin{pmatrix} (\mathbf{D}_1)_{11} & (\mathbf{D}_1)_{22} & \ldots & (\mathbf{D}_1)_{RR} \\ (\mathbf{D}_2)_{11} & (\mathbf{D}_2)_{22} & \ldots & (\mathbf{D}_2)_{RR} \\ & \vdots & & \\ (\mathbf{D}_L)_{11} & (\mathbf{D}_L)_{22} & \ldots & (\mathbf{D}_L)_{RR} \end{pmatrix}$$

are distinct.

Proof. Consider $Y = \mathbf{D}^T \cdot X$, for $X \in \mathbb{R}^L$. The ith and jth entry of Y are distinct if X is not perpendicular to $D_i - D_j$. Because $D_i \neq D_j$, the kernel of $D_i^T - D_j^T$ is a subspace of dimension $L-1$. Let $\mathbb{K}$ be the union of the kernels for all $i \neq j$ and let $\tilde{X} \in \mathbb{R}^L \setminus \mathbb{K}$. The EVD of $\sum_l \tilde{x}_l \mathbf{M}_l$ is given by

$$\sum_l \tilde{x}_l \mathbf{M}_l = \mathbf{U} \cdot \left(\sum_l \tilde{x}_l \mathbf{D}_l \right) \cdot \mathbf{U}^{-1} = \mathbf{U} \cdot \operatorname{diag}\{\mathbf{D}^T \cdot \tilde{X}\} \cdot \mathbf{U}^{-1}.$$

Because all eigenvalues are distinct, the eigenmatrix $\mathbf{U}$ is unique up to a permutation and a scaling of its columns. On the other hand, if columns of $\mathbf{D}$ are equal, it

is not possible to discriminate between different eigenvectors in the corresponding eigenspace. □

The equivalent for unitary diagonalization is given in [2].

Because of the link between (4.5)–(4.7) and (4.14)–(4.16), the CANDECOMP is essentially unique when $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are nonsingular and $\mathbf{U}^{(3)}$ does not contain collinear columns, as put forward in section 2.

Theorem 6.1 shows that a simultaneous EVD is much more robust than a single EVD. It is well known that, when eigenvalues are close, the eigenvectors in a single EVD may be strongly affected by small perturbations [30]. The reason is that for coinciding eigenvalues only the corresponding eigenspace is defined; different directions in this subspace will emerge as eigenvectors for different infinitesimal perturbations. When this happens for one or more of the matrices in a simultaneous EVD, the other matrices may still allow to identify the actual eigenvectors. We may conclude that, under the conditions of section 2, the CANDECOMP is likely to be stable.

Different permutations of the canonical components will correspond to entirely different matrices $\mathbf{Q}$ and $\mathbf{Z}$ in the SGSD (5.1)–(5.3). However, these in turn lead to different matrices $\mathbf{R}$ and $\mathbf{R}''$ such that, eventually, $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are still subject to the same indeterminacies. In other words, the uniqueness condition has not been weakened by formulating the problem in terms of orthogonal unknowns $\mathbf{Q}$, $\mathbf{Z}$.

It is worth mentioning that, for arbitrary matrices $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_K$ (not satisfying our CANDECOMP model), the uniqueness conditions of a S(G)SD are much more severe. In general, only one sequence of (generalized) Schur vectors is possible. For convenience, we will illustrate this only for the SSD (which, in our application, would arise from substitution of the QR-factorization of $\mathbf{U}^{(1)}$ in (4.14)–(4.16)). We have the following theorem.

THEOREM 6.2. *Let the matrices* $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_L \in \mathbb{R}^{R\times R}$ *satisfy the SSD*

$$\mathbf{M}_1 = \mathbf{Q}\cdot\mathbf{R}_1\cdot\mathbf{Q}^T, \tag{6.4}$$

$$\mathbf{M}_2 = \mathbf{Q}\cdot\mathbf{R}_2\cdot\mathbf{Q}^T, \tag{6.5}$$

$$\vdots$$

$$\mathbf{M}_L = \mathbf{Q}\cdot\mathbf{R}_L\cdot\mathbf{Q}^T, \tag{6.6}$$

with $\mathbf{Q} = [Q_1 Q_2 \ldots Q_R] \in \mathbb{R}^{R\times R}$ *orthogonal and* $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_L \in \mathbb{R}^{R\times R}$ *upper triangular. An equivalent simultaneous decomposition, in terms of* $\tilde{\mathbf{Q}}$ *and* $\{\tilde{\mathbf{R}}_l\}_{1\leqslant l\leqslant L}$, *in which the diagonal of* $(\tilde{\mathbf{R}}_l)$ *subsequently contains* $(\mathbf{R}_l)_{11}, (\mathbf{R}_l)_{22}, \ldots, (\mathbf{R}_l)_{I-1,I-1}$, $(\mathbf{R}_l)_{JJ}, (\mathbf{R}_l)_{I+1,I+1}, \ldots, (\mathbf{R}_l)_{J-1,J-1}, (\mathbf{R}_l)_{II}, (\mathbf{R}_l)_{J+1,J+1}, \ldots, (\mathbf{R}_l)_{KK}$ $(1 \leqslant l \leqslant L)$, *exists if and only if the following matrix is rank deficient*:

$$\begin{pmatrix} \mathbf{M}_1 - (\mathbf{R}_1)_{JJ}\,\mathbf{I} & [Q_1 \ldots Q_{I-1}] & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{M}_2 - (\mathbf{R}_2)_{JJ}\,\mathbf{I} & \mathbf{0} & [Q_1 \ldots Q_{I-1}] & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{M}_L - (\mathbf{R}_L)_{JJ}\,\mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & [Q_1 \ldots Q_{I-1}] \end{pmatrix}. \tag{6.7}$$

Proof. Let us first answer the simple question of which diagonal entry could be permuted to position $(1,1)$. There is a common eigenvector, other than Q_1, if and only if there exists a $J > 1$ such that all the equations

$$(\mathbf{M}_l - (\mathbf{R}_l)_{JJ}\,\mathbf{I})X = 0, \qquad 1 \leqslant l \leqslant L,$$

have a common solution. This is the condition specified by the theorem for $I = 1$. One can verify that the upper triangular structure can be maintained for new matrices $\tilde{\mathbf{R}}_1, \tilde{\mathbf{R}}_2, \ldots, \tilde{\mathbf{R}}_L$ and $\tilde{\mathbf{Q}}$ when the entries at position (J, J) are permuted to position $(1, 1)$ and the old entries at positions $(1, 1)$, $(2, 2)$, $\ldots$, $(J-1, J-1)$ are shifted one place down on the diagonal. (The strictly upper diagonal entries of rows 1 to J have to be recomputed.)

In general, the entries at position (J, J) can be brought in Ith position if and only if there exists a vector $X \neq 0$ and scalars b_{li}, $1 \leqslant l \leqslant L$, $1 \leqslant i \leqslant I-1$, such that

$$(\mathbf{M}_l - (\mathbf{R}_l)_{JJ}\,\mathbf{I})X = \sum_{i=1}^{I-1} b_{li}\,Q_i, \qquad 1 \leqslant l \leqslant L.$$

This is a set of homogeneous linear equations of which the unknowns are the coefficients of X and the scalars $\{b_{li}\}$. The coefficient matrix is given by (6.7). □

Moreover, for noisy data, different permutations of the canonical components will lead to matrices $\mathbf{Q}$, $\mathbf{Z}$ that yield different values of the cost function h defined in (5.6).

6.2. First-order perturbation analysis. To increase our understanding of the stability of the SGSD, let us now conduct a first-order perturbation analysis.

THEOREM 6.3. *Consider the function $g(\mathbf{Q}, \mathbf{Z})$ in (5.4) and let the matrices $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_K$ be defined by (5.1)–(5.3). The gradients of g, with respect to $\mathbf{Q}$ and $\mathbf{Z}$, over the manifold of orthogonal matrices, are given by*

$$\nabla_{\mathbf{Q}} g = 2\,\mathrm{skew}\left(\sum_k \mathrm{upp}(\mathbf{R}_k)\mathbf{R}_k^T\right)\cdot \mathbf{Q}, \tag{6.8}$$

$$\nabla_{\mathbf{Z}} g = 2\,\mathbf{Z}\cdot \mathrm{skew}\left(\sum_k \mathbf{R}_k^T \mathrm{upp}(\mathbf{R}_k)\right), \tag{6.9}$$

in which skew$(\cdot)$ *is the skew-symmetric and* upp$(\cdot)$ *the upper triangular part of a matrix.*

Proof. We will prove this result by resorting to the framework established in [15, 22]. The gradient of g with respect to $\mathbf{Q}$ can be determined by assuming that $\mathbf{Q}$ has a velocity $\dot{\mathbf{Q}}$ on the manifold of orthogonal matrices and expressing the evolution of g:

$$\dot{g} = \langle \nabla_{\mathbf{Q}}\, g, \dot{\mathbf{Q}} \rangle \tag{6.10}$$

(see, e.g., [15, p. 48]; the formula corresponds to a chain rule for the derivation).

First we express the function g as

$$g(\mathbf{Q}, \mathbf{Z}) = \sum_{k=1}^{K} \langle \mathbf{Q}\cdot\mathbf{V}_k\cdot\mathbf{Z}, \mathrm{upp}(\mathbf{Q}\cdot\mathbf{V}_k\cdot\mathbf{Z})\rangle.$$

Assuming that $\mathbf{Q}$ is time dependent, the derivative with respect to the time coordinate is given by (taking into account that upp$(\cdot)$ is a linear operation)

$$\begin{aligned}\dot{g} &= \sum_{k=1}^{K} [\langle \dot{\mathbf{Q}}\cdot\mathbf{V}_k\cdot\mathbf{Z}, \mathrm{upp}(\mathbf{Q}\cdot\mathbf{V}_k\cdot\mathbf{Z})\rangle + \langle \mathbf{Q}\cdot\mathbf{V}_k\cdot\mathbf{Z}, \mathrm{upp}(\dot{\mathbf{Q}}\cdot\mathbf{V}_k\cdot\mathbf{Z})\rangle] \\ &= 2\sum_{k=1}^{K} \langle \dot{\mathbf{Q}}\cdot\mathbf{V}_k\cdot\mathbf{Z}, \mathrm{upp}(\mathbf{Q}\cdot\mathbf{V}_k\cdot\mathbf{Z})\rangle.\end{aligned}$$

With a property of the scalar product, we obtain

$$\dot{g} = 2 \sum_{k=1}^{K} \langle \dot{\mathbf{Q}}, \operatorname{upp}(\mathbf{Q} \cdot \mathbf{V}_k \cdot \mathbf{Z}) \cdot \mathbf{Z}^T \cdot \mathbf{V}_k^T \rangle.$$

The right term is proportional to the gradient of g over $\mathbb{R}^{R \times R}$. To ensure that $\mathbf{Q}$ stays on the manifold of orthogonal matrices, we claim additionally that

$$\dot{\mathbf{Q}} = \boldsymbol{\Omega} \cdot \mathbf{Q},$$

in which $\boldsymbol{\Omega} \in \mathbb{R}^{R \times R}$ is skew-symmetric [22, p. 307]. Now the inner product can be written in the form of (6.10):

$$\begin{aligned}
\dot{g} &= 2 \sum_{k=1}^{K} \langle \boldsymbol{\Omega}, \operatorname{upp}(\mathbf{Q} \cdot \mathbf{V}_k \cdot \mathbf{Z}) \cdot \mathbf{Z}^T \cdot \mathbf{V}_k^T \cdot \mathbf{Q}^T \rangle \\
&= \left\langle \boldsymbol{\Omega}, 2 \sum_{k=1}^{K} \operatorname{skew}\{\operatorname{upp}(\mathbf{Q} \cdot \mathbf{V}_k \cdot \mathbf{Z}) \cdot \mathbf{Z}^T \cdot \mathbf{V}_k^T \cdot \mathbf{Q}^T\} \right\rangle \\
&= \left\langle \boldsymbol{\Omega} \cdot \mathbf{Q}, 2 \sum_{k=1}^{K} \operatorname{skew}\{\operatorname{upp}(\mathbf{Q} \cdot \mathbf{V}_k \cdot \mathbf{Z}) \cdot \mathbf{Z}^T \cdot \mathbf{V}_k^T \cdot \mathbf{Q}^T\} \cdot \mathbf{Q} \right\rangle,
\end{aligned}$$

which proves (6.8). The gradient with respect to $\mathbf{Z}$ can be found in an analogous way. ∎

THEOREM 6.4. *Consider a first-order perturbation of the matrices in the SGSD* (5.1)–(5.3): $\mathbf{V}_k(\epsilon) = \mathbf{V}_k(0) + \epsilon \mathbf{B}_k$ $(1 \leqslant k \leqslant K)$. *As a first-order approximation, the maximum of* $g(\mathbf{Q}, \mathbf{Z})$ *is then obtained for*

$$\begin{aligned}
\mathbf{Q}(\epsilon) &= (\mathbf{I} + \epsilon \boldsymbol{\Lambda} + o(\epsilon)) \cdot \mathbf{Q}(0), \\
\mathbf{Z}(\epsilon) &= \mathbf{Z}(0) \cdot (\mathbf{I} + \epsilon \boldsymbol{\Omega} + o(\epsilon)),
\end{aligned}$$

in which $\boldsymbol{\Lambda}, \boldsymbol{\Omega} \in \mathbb{R}^{R \times R}$ *are skew-symmetric matrices that satisfy the following set of linear equations*:

$$\sum_k \operatorname{lows}(\mathbf{R}_k \boldsymbol{\Omega} + \mathbf{E}_k + \boldsymbol{\Lambda} \mathbf{R}_k) \cdot \mathbf{R}_k^T = \mathbf{0}, \tag{6.11}$$

$$\sum_k \mathbf{R}_k^T \cdot \operatorname{lows}(\mathbf{R}_k \boldsymbol{\Omega} + \mathbf{E}_k + \boldsymbol{\Lambda} \mathbf{R}_k) = \mathbf{0}, \tag{6.12}$$

where $\operatorname{lows}(\cdot)$ *is the strictly lower triangular part of a matrix and*

$$\mathbf{E}_k = \mathbf{Q}(0) \cdot \mathbf{B}_k \cdot \mathbf{Z}(0), \qquad 1 \leqslant k \leqslant K.$$

Proof. Again, we will work in the framework of [15, 22]. Let us start from (5.1)–(5.3). If the matrices $\mathbf{A}_k$ have a velocity $\dot{\mathbf{A}}_k = \mathbf{B}_k$, then $\mathbf{Q}$ evolves in such a way that the identity $\nabla_{\mathbf{Q}} g \equiv 0$ holds. Taking the form of the gradient (6.8) into account, we should have that

$$\operatorname{skew}\left(\sum_k \operatorname{upp}(\mathbf{R}_k) \mathbf{R}_k^T \right) \equiv 0. \tag{6.13}$$

Taking the derivative with respect to the time coordinate yields

$$\text{skew}\left(\sum_k \text{upp}(\dot{\mathbf{Q}}\cdot\mathbf{A}_k\cdot\mathbf{Z}+\mathbf{Q}\cdot\dot{\mathbf{A}}_k\cdot\mathbf{Z}+\mathbf{Q}\cdot\mathbf{A}_k\cdot\dot{\mathbf{Z}})\mathbf{R}_k^T \right.$$
$$\left. +\text{upp}(\mathbf{R}_k)(\dot{\mathbf{Z}}^T\cdot\mathbf{A}_k^T\cdot\mathbf{Q}^T+\mathbf{Z}^T\cdot\dot{\mathbf{A}}_k^T\cdot\mathbf{Q}^T+\mathbf{Z}^T\cdot\mathbf{A}_k^T\cdot\dot{\mathbf{Q}}^T)\right)=0.$$

To ensure that $\mathbf{Q}$ and $\mathbf{Z}$ stay on the manifold of orthogonal matrices, we claim that

$$\dot{\mathbf{Q}}=\mathbf{\Omega}\cdot\mathbf{Q},$$
$$\dot{\mathbf{Z}}=\mathbf{Z}\cdot\mathbf{\Lambda},$$

in which $\mathbf{\Omega},\mathbf{\Lambda}\in\mathbb{R}^{R\times R}$ are skew-symmetric. If (5.1)–(5.3) are exactly satisfied, then $\text{upp}(\mathbf{R}_k)=\mathbf{R}_k$. Substitution of $\mathbf{E}_k=\mathbf{Q}\cdot\mathbf{B}_k\cdot\mathbf{Z}$ then yields

$$\text{skew}\left(\sum_k \mathbf{R}_k\cdot\mathbf{\Omega}\cdot\mathbf{R}_k^T-\text{upp}(\mathbf{R}_k\cdot\mathbf{\Omega})\cdot\mathbf{R}_k^T+\mathbf{E}_k\cdot\mathbf{R}_k^T-\text{upp}(\mathbf{E}_k)\cdot\mathbf{R}_k^T\right.$$
$$\left.+\mathbf{\Lambda}\cdot\mathbf{R}_k\cdot\mathbf{R}_k^T-\text{upp}(\mathbf{\Lambda}\cdot\mathbf{R}_k)\cdot\mathbf{R}_k^T\right)=0$$

or

$$\text{skew}\left(\sum_k \text{lows}(\mathbf{R}_k\mathbf{\Omega}+\mathbf{E}_k+\mathbf{\Lambda}\mathbf{R}_k)\mathbf{R}_k^T\right)=0.$$

We may drop "skew" because its argument is strictly lower triangular. Equation (6.12) is obtained by starting from the identity $\nabla_{\mathbf{Z}}g\equiv 0$. □

Remark 7. For matrices $\mathbf{A}_k$ that do not allow for an exact upper triangularization, the derivation can be taken over provided that $\text{upp}(\mathbf{R}_k)$ is not simplified to $\mathbf{R}_k$.

Remark 8. Note that the expressions derived in this section may be used to develop routines for the computation of the SGSD by means of an optimization over the (product of two) manifold(s) of orthogonal matrices. We refer to [22].

By the summation in (6.11) and (6.12) the perturbation is to some extent "averaged" over the different matrices $\mathbf{A}_k$. When components of $\mathbf{Q}$ and $\mathbf{Z}$ are ill conditioned for a subset of $\{\mathbf{A}_k\}$, this may be compensated by the other matrices.

7. Algorithms for the SGSD.

7.1. Extended QZ-iteration. For the actual computation of the SGSD, an extended QZ-iteration was proposed in [48]. One alternates between updates of $\mathbf{Q}$ and $\mathbf{Z}$ in such a way that the cost function h in (5.6) is approximately optimized. In each step, the estimate of $\mathbf{Q}$ (given $\mathbf{Z}$, or vice-versa) is obtained as a product of matrices $\mathbf{H}_1\mathbf{H}_2\ldots\mathbf{H}_{R-1}$, that form the equivalent of Householder matrices for the computation of a simple QR-decomposition [24]. For instance, as far as $\mathbf{Q}$ is concerned, $\mathbf{H}_1$ maximally reduces (in least-squares sense) the below-diagonal norm of the first columns of the instantaneous estimates of $\mathbf{R}_1,\mathbf{R}_2,\ldots,\mathbf{R}_K$. After multiplication with $\mathbf{H}_1$, $\mathbf{H}_2$ minimizes the below-diagonal norm of the second columns, without further affecting the first rows, and so on. $\mathbf{H}_1$ is determined through an SVD of an $(R\times K)$-matrix (actually only the left singular vector corresponding to the largest singular value, and

its orthogonal complement, have to be computed), the determination of $\mathbf{H}_2$ involves an SVD of an $((R-1)\times K)$-matrix, and so on.

Because of the high computational cost, it makes sense to initialize the algorithm with matrices $\mathbf{Q}^{(0)}$ and $\mathbf{Z}^{(0)}$ defined by two of the equations (5.1)–(5.3). If these two joint decompositions are well conditioned, then $\mathbf{Q}^{(0)}$ and $\mathbf{Z}^{(0)}$ may be close to the optimum; if not, then the extended QZ-iteration may involve more work than just a fine tuning of a good initialization.

The resulting scheme is observed to find a good estimate of the global optimum in a limited number of steps, if the CANDECOMP-model is exactly satisfied. However, even moderate perturbations can cause the algorithm to end up in good estimates of the theoretical matrices $\mathbf{Q}$ and $\mathbf{Z}$ that do not globally minimize the cost function. It is also possible that at some point in the iteration (e.g., initially, or after approximate convergence), the algorithm starts to increase the value of h. The reason for this behavior is that the way in which $\mathbf{Q}$ and $\mathbf{Z}$ are computed does not imply monotonic convergence in terms of h: for instance, it is possible that the matrix $\mathbf{H}_1$ increases the Frobenius-norm of the part of columns 2 to $R-1$ below the diagonal. Nevertheless, these aspects do not seem to pose major problems in practice: over several hundreds of simulations, we have only once obtained a meaningless result.

7.2. Jacobi iteration. In [17, 18] we derived a Jacobi-type algorithm for the computation of the SGSD. Here, $\mathbf{Q}$ and $\mathbf{Z}$ are found as a sequence of elementary Jacobi-rotation matrices. In a step (i,j), $\mathbf{Q}$ and $\mathbf{Z}$ are multiplied by elementary rotation matrices, affecting rows and columns i and j. These rotation matrices are such that they maximize the function g in (5.4). It turns out that the determination of a Jacobi-rotation pair basically amounts to rooting a polynomial of degree 8. One sweeps over all the possible pairs (i,j), and then iterates over such sweeps.

The iteration can be initialized with matrices $\mathbf{Q}^{(0)}$ and $\mathbf{Z}^{(0)}$, obtained from the generalized Schur decomposition corresponding to two of the equations in (5.1)–(5.3) [24]. Assume that at iteration step $l+1$, the estimates $\mathbf{Q}^{(l)}$, $\mathbf{Z}^{(l)}$, and $\mathbf{R}_1^{(l)},\ldots,\mathbf{R}_K^{(l)}$ are available. Let $\mathbf{G}_{ij}\in\mathbb{R}^{R\times R}$ represent an elementary Givens rotation matrix that affects rows i and j, i.e., $\mathbf{G}_{ij}$ equals the identity matrix, except for the entries

$$(\mathbf{G}_{ij})_{ii} = (\mathbf{G}_{ij})_{jj} = \cos\alpha,$$
$$(\mathbf{G}_{ij})_{ji} = -(\mathbf{G}_{ij})_{ij} = \sin\alpha,$$

in which α is the rotation angle (assume that $j>i$). An update of $\mathbf{Q}^{(l)}$ takes the form of $\mathbf{Q}^{(l+1)} = \mathbf{G}_{ij}\cdot\mathbf{Q}^{(l)}$. Similarly, an update of $\mathbf{Z}^{(l)}$ takes the form of $\mathbf{Z}^{(l+1)} = \mathbf{Z}^{(l)}\cdot{\mathbf{G}'_{ij}}^{T}$, where the Givens rotation matrix $\mathbf{G}'_{ij}$ is defined in the same way as $\mathbf{G}_{ij}$, in terms of an angle β. At the same time $\mathbf{R}_1^{(l)},\mathbf{R}_2^{(l)},\ldots,\mathbf{R}_K^{(l)}$ are updated as $\mathbf{R}_1^{(l+1)} = \mathbf{G}_{ij}\cdot\mathbf{R}_1^{(l)}\cdot{\mathbf{G}'_{ij}}^{T}$, $\mathbf{R}_2^{(l+1)} = \mathbf{G}_{ij}\cdot\mathbf{R}_2^{(l)}\cdot{\mathbf{G}'_{ij}}^{T},\ldots,\mathbf{R}_K^{(l+1)} = \mathbf{G}_{ij}\cdot\mathbf{R}_{\mathbf{K}}{}^{(l)}\cdot{\mathbf{G}'_{ij}}^{T}$.

At iteration step l, the maximization of the function g in (5.4) is equivalent to the minimization of

$$h(\alpha,\beta) = \sum_{k=1}^{K}\left[(\mathbf{R}_k^{(l+1)})_{ji}^2 + \sum_{r=i+1}^{j-1}((\mathbf{R}_k^{(l+1)})_{ri}^2 + (\mathbf{R}_k^{(l+1)})_{jr}^2)\right] \tag{7.1}$$

(the other entries do not affect the norm of the strictly lower diagonal parts). The function h is given in explicit form by

$$h(\alpha,\beta) = \sum_{k=1}^{K}\sum_{n=1}^{5} h_{kn}(\alpha,\beta), \tag{7.2}$$

in which

$$h_{k1}(\alpha,\beta) = \sin^2\alpha$$
$$\times[\cos^2\beta\,(\mathbf{R}_k^{(l)})_{ii}^2 + \sin^2\beta\,(\mathbf{R}_k^{(l)})_{ij}^2 - 2\sin\beta\cos\beta\,(\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{ij}], \tag{7.3}$$

$$h_{k2}(\alpha,\beta) = 2\sin\alpha\cos\alpha\,\Big\{\cos^2\beta\,(\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{ji} + \sin^2\beta\,(\mathbf{R}_k^{(l)})_{ij}(\mathbf{R}_k^{(l)})_{jj}$$
$$- \sin\beta\cos\beta\,[(\mathbf{R}_k^{(l)})_{ij}(\mathbf{R}_k^{(l)})_{ji} + (\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{jj}]\Big\}, \tag{7.4}$$

$$h_{k3}(\alpha,\beta) = \cos^2\alpha$$
$$\times[\cos^2\beta\,(\mathbf{R}_k^{(l)})_{ji}^2 + \sin^2\beta\,(\mathbf{R}_k^{(l)})_{jj}^2 - 2\sin\beta\cos\beta\,(\mathbf{R}_k^{(l)})_{ji}(\mathbf{R}_k^{(l)})_{jj}], \tag{7.5}$$

$$h_{k4}(\alpha,\beta) = (\sin^2\alpha + \cos^2\alpha)$$
$$\times \sum_{r=i+1}^{j-1}[\cos^2\beta\,(\mathbf{R}_k^{(l)})_{ri}^2 + \sin^2\beta\,(\mathbf{R}_k^{(l)})_{rj}^2 - 2\sin\beta\cos\beta\,(\mathbf{R}_k^{(l)})_{ri}(\mathbf{R}_k^{(l)})_{rj}], \tag{7.6}$$

$$h_{k5}(\alpha,\beta) = (\sin^2\beta + \cos^2\beta)$$
$$\times \sum_{r=i+1}^{j-1}[\cos^2\alpha\,(\mathbf{R}_k^{(l)})_{jr}^2 + \sin^2\alpha\,(\mathbf{R}_k^{(l)})_{ir}^2 + 2\sin\alpha\cos\alpha\,(\mathbf{R}_k^{(l)})_{ir}(\mathbf{R}_k^{(l)})_{jr}]. \tag{7.7}$$

Setting the partial derivatives of h, with respect to α and β, equal to zero, leads to a set of biquadratic equations in $\tan\alpha$ and $\tan\beta$:

$$b_1(\beta)\tan^2\alpha + b_2(\beta)\tan\alpha - b_1(\beta) = 0, \tag{7.8}$$

$$b_3(\beta)\tan^2\alpha + b_4(\beta)\tan\alpha + b_5(\beta) = 0, \tag{7.9}$$

in which $b_n(\beta) = \sum_{k=1}^{K} b_{kn}(\beta)$, with

$$b_{k1}(\beta) = \tan^2\beta\,\left\{(\mathbf{R}_k^{(l)})_{ij}^2 - (\mathbf{R}_k^{(l)})_{jj}^2 + \sum_{r=i+1}^{j-1}[(\mathbf{R}_k^{(l)})_{ir}^2 - (\mathbf{R}_k^{(l)})_{jr}^2]\right\}$$
$$+2\tan\beta\,[(\mathbf{R}_k^{(l)})_{ji}(\mathbf{R}_k^{(l)})_{jj} - (\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{ij}]$$
$$+\left\{(\mathbf{R}_k^{(l)})_{ii}^2 - (\mathbf{R}_k^{(l)})_{ji}^2 + \sum_{r=i+1}^{j-1}[(\mathbf{R}_k^{(l)})_{ir}^2 - (\mathbf{R}_k^{(l)})_{jr}^2]\right\}, \tag{7.10}$$

$$b_{k2}(\beta) = \tan^2\beta\,\left[(\mathbf{R}_k^{(l)})_{ij}(\mathbf{R}_k^{(l)})_{jj} + \sum_{r=i+1}^{j-1}(\mathbf{R}_k^{(l)})_{ir}(\mathbf{R}_k^{(l)})_{jr}\right]$$
$$-\tan\beta\,[(\mathbf{R}_k^{(l)})_{ij}(\mathbf{R}_k^{(l)})_{ji} + (\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{jj}]$$
$$+\left[(\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{ji} + \sum_{r=i+1}^{j-1}(\mathbf{R}_k^{(l)})_{ir}(\mathbf{R}_k^{(l)})_{jr}\right], \tag{7.11}$$

$$b_{k3}(\beta) = (\tan^2\beta - 1)\,\left[(\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{ij} + \sum_{r=i+1}^{j-1}(\mathbf{R}_k^{(l)})_{ri}(\mathbf{R}_k^{(l)})_{jr}\right]$$
$$+\tan\beta\,\left\{(\mathbf{R}_k^{(l)})_{ij}^2 - (\mathbf{R}_k^{(l)})_{ii}^2 + \sum_{r=i+1}^{j-1}[(\mathbf{R}_k^{(l)})_{rj}^2 - (\mathbf{R}_k^{(l)})_{ri}^2]\right\}, \tag{7.12}$$

$$b_{k4}(\beta) = (\tan^2\beta - 1)\,[(\mathbf{R}_k^{(l)})_{ij}(\mathbf{R}_k^{(l)})_{ji} + (\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{jj}]$$

$$+2\tan\beta\,[(\mathbf{R}_k^{(l)})_{ij}(\mathbf{R}_k^{(l)})_{jj} - (\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{ji}], \tag{7.13}$$

$$b_{k5}(\beta) = (\tan^2\beta - 1)\left[(\mathbf{R}_k^{(l)})_{ji}(\mathbf{R}_k^{(l)})_{jj} + \sum_{r=i+1}^{j-1}(\mathbf{R}_k^{(l)})_{ri}(\mathbf{R}_k^{(l)})_{rj}\right]$$

$$+\tan\beta\left\{(\mathbf{R}_k^{(l)})_{jj}^2 - (\mathbf{R}_k^{(l)})_{ji}^2 + \sum_{r=i+1}^{j-1}[(\mathbf{R}_k^{(l)})_{rj}^2 - (\mathbf{R}_k^{(l)})_{ri}^2]\right\}. \tag{7.14}$$

The global minimum of $h(\alpha,\beta)$ can be determined by computing the various solutions of (7.8)–(7.9) and selecting the one corresponding to the smallest value in (7.2).

For the solution of the set of biquadratic equations, let us first consider the special case where (7.8) is linear in $\tan\alpha$: $b_1(\beta) = 0$. The only ways in which a root β_0 of b_1 can lead to a solution of (7.8)–(7.9) are (a) $\tan\alpha = 0$ and additionally $b_5(\beta_0) = 0$ and (b) α is a solution of (7.9), for $\beta = \beta_0$, which additionally satisfies $b_2(\beta_0) = 0$.

Now let us investigate the general case, i.e., $b_1(\beta) \neq 0$. Substitution of the square roots of (7.8) in (7.9) (considered as quadratic expressions in the unknown $\tan\alpha$) then leads to the following polynomial of degree 8 in $\tan\beta$:

$$\begin{aligned} &b_1^2(\beta)b_3^2(\beta) + b_1^2(\beta)b_5^2(\beta) - b_1(\beta)b_2(\beta)b_4(\beta)b_5(\beta) + 2b_1^2(\beta)b_3(\beta)b_5(\beta) \\ &+ b_2^2(\beta)b_3(\beta)b_5(\beta) - b_1^2(\beta)b_4^2(\beta) + b_1(\beta)b_2(\beta)b_3(\beta)b_4(\beta) = 0. \end{aligned} \tag{7.15}$$

For the roots of this polynomial, the corresponding value of $\tan\alpha$ that gives a solution to (7.8)–(7.9), can be found from

$$(b_2(\beta)b_3(\beta) - b_1(\beta)b_4(\beta))\tan\alpha - b_1(\beta)(b_3(\beta) + b_5(\beta)) = 0. \tag{7.16}$$

The computational cost is in line with results obtained for other simultaneous matrix decompositions. A Jacobi-rotation for a simultaneous real symmetric EVD can be computed by rooting a polynomial of degree 2 [8, 9]. For an SSD ($\mathbf{Q} = \mathbf{Z}$), polynomials are of degree 4 [25].

The Jacobi-result is an explicit solution for the CANDECOMP of rank-2 tensors. Apart from this result, a Jacobi-sweep is more expensive than an extended QZ-step if not $\min\{R, K\} \gg 8$.

If the simultaneous equivalence transformation of (4.5)–(4.7) is not exactly satisfied, different permutations of the CANDECOMP components may cause the corresponding orthogonal factors $\mathbf{Q}$ and $\mathbf{Z}$ to yield values of the function g that are somewhat different. There is no guarantee that the Jacobi-algorithm will converge to the solution with that specific column ordering that leads to the global optimum. Apart from the reordering of columns, there is no formal evidence that the two-sided Jacobi-algorithm cannot get stuck in a local optimum; local or global convergence is still an open problem for the computation of other simultaneous matrix decompositions as well [4, 8, 9, 12, 23, 48]. We have not observed convergence to a local optimum in any of our simulations for the unsymmetric CANDECOMP-problem. For the case where $\mathbf{U}^{(1)} = \mathbf{U}^{(2)}$, a meaningless result has been obtained for one out of hundreds of simulations. In this odd case, the problem could be overcome by reinitializing the algorithm.

8. Estimation of the canonical components from the components of the SGSD. In this section we will explain how the matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ can be estimated, once $\mathbf{Q}$ and $\mathbf{Z}$ are known. How $\mathbf{U}^{(3)}$ may subsequently be estimated was explained in section 4. This corresponds to step 4 in Algorithm 1. Computation of

the SGSD is in general only equivalent to least-squares fitting of the CANDECOMP-model if that model is exactly valid. The estimates obtained so far may then be used to initialize an additional optimization algorithm for the minimization of cost function (2.3), as also mentioned in section 4 (step 5 in Algorithm 1).

In [48] a procedure has been proposed that works under the assumption that the columns of $\mathbf{U}^{(3)}$ are linearly independent (and sufficiently well conditioned). Hence this technique can be used only when $K \geqslant R$. The solution is obtained via the computation of the pseudoinverse of a $(K \times R)$ matrix and the estimation of the best rank-1 approximation of R $(R \times R)$ matrices.

We will derive a new technique that works under the assumptions established in section 2. This technique is also computationally less demanding. It essentially requires solving $R(R-1)/2$ overdetermined sets of K linear equations in 2 unknowns.

We will estimate $\mathbf{R}'$ and $\mathbf{R}''$ from (5.1)–(5.3) and then combine them with $\mathbf{Q}$ and $\mathbf{Z}$ to obtain $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$. If we assume that the main diagonals of $\mathbf{R}'$ and $\mathbf{R}''$ contain only entries equal to 1 (we can make this assumption because the columns of $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ can be determined only up to a scaling factor), then $\mathbf{D}_k = \text{diag}\{\mathbf{R}_k\}$ $(1 \leqslant k \leqslant K)$, in which $\text{diag}\{\cdot\}$ now denotes the diagonal part of a matrix. The strictly upper diagonal elements of $\mathbf{R}'$ and $\mathbf{R}''$ can be estimated by subsequently solving in a least-squares sense the equations related to the entries of $\{\mathbf{R}_k\}_{(1\leqslant k\leqslant K)}$ at positions $(R-1,R)$, $(R-2,R-1)$, $(R-2,R)$, $\ldots$, $(1,2)$, $(1,3)$, $\ldots$, $(1,R)$ in (5.1)–(5.3) with respect to the unknowns $r'_{R-1,R}$ and $r''_{R-1,R}$, $r'_{R-2,R-1}$ and $r''_{R-2,R-1}$, $r'_{R-2,R}$, and $r''_{R-2,R}$, $\ldots$, $r'_{1,2}$ and $r''_{1,2}$, $r'_{1,3}$ and $r''_{1,3}$, $\ldots$, $r'_{1,R}$ and $r''_{1,R}$, respectively. For instance, with the entries at position $(R-1,R)$ corresponds the equation

$$\begin{pmatrix} (\mathbf{R}_1)_{R,R} & (\mathbf{R}_1)_{R-1,R-1} \\ (\mathbf{R}_2)_{R,R} & (\mathbf{R}_2)_{R-1,R-1} \\ \vdots & \vdots \\ (\mathbf{R}_K)_{R,R} & (\mathbf{R}_K)_{R-1,R-1} \end{pmatrix} \begin{pmatrix} r'_{R-1,R} \\ r''_{R-1,R} \end{pmatrix} = \begin{pmatrix} (\mathbf{R}_1)_{R-1,R} \\ (\mathbf{R}_2)_{R-1,R} \\ \vdots \\ (\mathbf{R}_K)_{R-1,R} \end{pmatrix}.$$

Note that, according to the third working assumption made in section 2, the columns of the matrix on the left-hand side of this equation should be linearly independent.

For the computation of $\mathbf{U}^{(3)}$, remark that (4.5)–(4.7) correspond to a CANDECOMP of a tensor $\mathcal{V} \in \mathbb{R}^{R\times R\times K}$, with entries $v_{ijk} = (\mathbf{V}_k)_{ij}$, of which the component matrices are $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$ and the matrix $\tilde{\mathbf{U}}^{(3)}$ defined in (4.8). Let $\mathbf{V}_{(R^2\times K)} \in \mathbb{R}^{R^2\times K}$, with entries

$$(\mathbf{V}_{(R^2\times K)})_{(i-1)R+j,k} = (\mathbf{V}_k)_{ij},$$

be a matrix representation of $\mathcal{V}$. (4.5)–(4.7) can be reformulated as

$$\mathbf{V}_{(R^2\times K)} = (\mathbf{U}^{(1)} \odot \mathbf{U}^{(2)}) \cdot \tilde{\mathbf{U}}^{(3)^T}. \tag{8.1}$$

$\tilde{\mathbf{U}}^{(3)}$ can be computed from this (possibly overdetermined) set of linear equations. Finally, $\mathbf{U}^{(3)}$ follows from (4.9).

To conclude, let us give an outline of the computation of $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, and $\mathbf{U}^{(3)}$ from the results of the SGSD (5.1)–(5.3). This scheme details step 4 in Algorithm 1.

4.1 Computation of $\mathbf{R}'$ and $\mathbf{R}''$.
Set $\text{diag}\{\mathbf{R}'\} = \text{diag}\{\mathbf{R}''\} = \mathbf{I}$.
for $i = R-1, R-2, \ldots, 1$

for $j = i+1, i+2, \ldots, R$

$$\begin{pmatrix} (\mathbf{R}_1)_{jj} & (\mathbf{R}_1)_{ii} \\ (\mathbf{R}_2)_{jj} & (\mathbf{R}_2)_{ii} \\ \vdots & \vdots \\ (\mathbf{R}_K)_{jj} & (\mathbf{R}_K)_{ii} \end{pmatrix} \begin{pmatrix} r'_{ij} \\ r''_{ij} \end{pmatrix} = \begin{pmatrix} (\mathbf{R}_1)_{ij} - \sum_{p=i+1}^{j-1} r'_{ip} (\mathbf{R}_1)_{pp} r''_{pj} \\ (\mathbf{R}_2)_{ij} - \sum_{p=i+1}^{j-1} r'_{ip} (\mathbf{R}_2)_{pp} r''_{pj} \\ \vdots \\ (\mathbf{R}_K)_{ij} - \sum_{p=i+1}^{j-1} r'_{ip} (\mathbf{R}_K)_{pp} r''_{pj} \end{pmatrix}$$

end

end

4.2 $\mathbf{U}^{(1)} = \mathbf{Q}^T \cdot \mathbf{R}'$. $\mathbf{U}^{(2)} = \mathbf{Z} \cdot (\mathbf{R}'')^T$.

4.3 Compute $\tilde{\mathbf{U}}^{(3)}$ from (8.1). Compute $\mathbf{U}^{(3)}$, modulo a scaling of its columns, from (4.9).

9. Numerical experiments. In this section we illustrate the performance of the algorithms proposed in this paper by means of a number of numerical experiments. These experiments are helpful to understand and evaluate the different methods, given that a rigorous mathematical analysis of their convergence properties often proves to be extremely tough (as is witnessed by the fact that only very few related results are available [4, 50]).

In a first series of experiments we will compare the accuracy of both techniques presented in section 7 and check whether an additional direct optimization of the cost function f, defined in (2.3), is needed (step 5 in Algorithm 1). We will also show that the extended QZ-iteration is not simply based on the minimization of cost function h, defined in (5.6).

Tensors $\mathcal{A} \in \mathbb{R}^{3\times3\times3}$, of which the canonical components will afterwards be estimated, are generated in the following way:

$$\mathcal{A} = \tilde{\mathcal{A}}/\|\tilde{\mathcal{A}}\| + \sigma_N \tilde{\mathcal{N}}/\|\tilde{\mathcal{N}}\|, \tag{9.1}$$

in which $\tilde{\mathcal{A}}$ exactly satisfies the CANDECOMP-model:

$$\tilde{\mathcal{A}} = U_1^{(1)} \circ U_1^{(2)} \circ U_1^{(3)} + U_2^{(1)} \circ U_2^{(2)} \circ U_2^{(3)} + U_3^{(1)} \circ U_3^{(2)} \circ U_3^{(3)}. \tag{9.2}$$

The components in (9.1)–(9.2) are generated as follows. First consider the (3×3)-matrices $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, and $\mathbf{U}^{(3)}$, defined by (4.3). The entries of 3 (3×3)-matrices are randomly taken from a uniform distribution on the interval $[0, 1)$. $\mathbf{U}^{(2)}$ and $\mathbf{U}^{(3)}$ are derived from two of these matrices by replacing their singular values by 3, 2, 1, while keeping the singular vectors. $\mathbf{U}^{(1)}$ is generated in the same way but three different sets of singular values will be considered: 3, 2, 1; 30, 15, 1; 100, 50, 1. The entries of $\tilde{\mathcal{N}}$ are drawn from a zero-mean unit-variance Gaussian distribution. For each particular choice of $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, $\mathbf{U}^{(3)}$, and $\tilde{\mathcal{N}}$, the scalar σ_N is varied between $1e-3$ and 1.

For each of the sets of singular values of $\mathbf{U}^{(1)}$, 50 independent samples of $\tilde{\mathcal{A}}$ are realized; for each of them 7 logarithmically equidistant values of σ_N are considered. In each Monte Carlo simulation the following algorithms are run: (a) the Jacobi-algorithm, discussed in section 7.2; (b) a least-squares matching of both sides of (2.3), for which the `leastsq` command of the Optimization Toolbox 1.0 of MATLAB 4.2 has been used, initialized with the result of (a); and (c) the extended QZ-iteration, described in section 7.1. The algorithm (a) is terminated if a full sweep no longer allows the reduction of the cost function $h(\mathbf{Q}, \mathbf{Z})$ with at least 0.01%. The same termination criterion is used for a Q-step followed by a Z-step in the extended QZ-iteration. For

the least-squares matching (b) a minimal precision of $1e-5$ for the optimal values of the cost function f, defined in (2.3), and the corresponding components is presumed; the MATLAB routine maximally performs 2100 iteration steps.

To evaluate the accuracy of the different algorithms we will consider the quality of the estimate $\hat{\mathbf{U}}^{(1)}$ of $\mathbf{U}^{(1)}$; at this point the columns of $\mathbf{U}^{(1)}$ are normalized to unit-length. In Figure 9.1 the error is plotted as a function of the noise level σ_N. For a given noise level and a given algorithm, this error measure has been computed as the average, over the different Monte Carlo simulations, of the Frobenius-norm $\|\hat{\mathbf{U}}^{(1)} - \mathbf{U}^{(1)}\|$; the ordering of the columns of $\hat{\mathbf{U}}^{(1)}$ that corresponds to the ordering of the columns of $\mathbf{U}^{(1)}$, has been determined as the ordering that minimizes the error; we also scale the columns of $\hat{\mathbf{U}}^{(1)}$ in the optimal way. Algorithms (a), (b), and (c) correspond to solid, dotted, and dash-dot curves, respectively. The upper, middle, and lower curves correspond to a condition number of $\mathbf{U}^{(1)}$, equal to 100, 30 and 3, respectively.

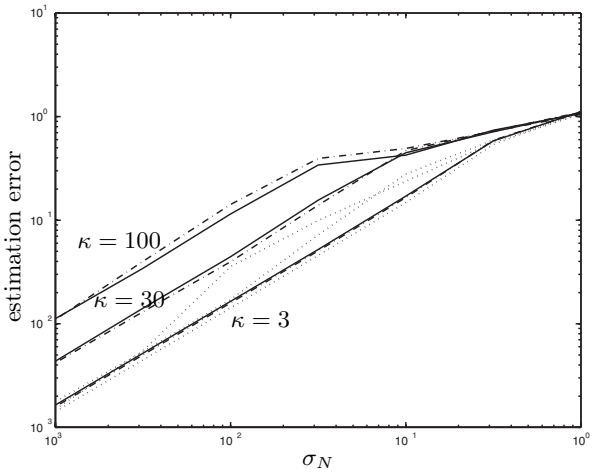


FIG. 9.1. *The mean value of* $\|\hat{\mathbf{U}}^{(1)} - \mathbf{U}^{(1)}\|$, *as a function of the noise level* σ_N, *for the Jacobi-algorithm (solid), with additional least-squares matching (dotted) and the extended QZ-algorithm (dashdot). The upper, middle, and lower curves correspond to a condition number* κ *of* $\mathbf{U}^{(1)}$, *equal to* 100, 30, *and* 3, *respectively.*

Figure 9.1 displays the expected performance degradation as the noise level and/or the condition number of $\mathbf{U}^{(1)}$ increases. The number of simulations is high enough to give a good picture: the variance of the error, divided by its squared value ranges from at least $4e-6$ ($\sigma_N = 1e-3$) to typically $2.5e-2$ ($\sigma_N = 1$). We notice that on the average, the accuracy of methods (a) and (c) is comparable. The figure also shows that an additional least-squares matching routine generally improved the accuracy, but that the marginal improvement became smaller as the CANDECOMP factors were better conditioned. For well-conditioned problems, no direct optimization of f is needed.

In Figure 9.2 we have plotted the mean value of the cost function h in (5.6) for the algorithms (a) and (c). The figure shows that the extended QZ-iteration indeed does not minimize cost function h; this effect is more outspoken as the condition number of $\mathbf{U}^{(1)}$ is larger. On the other hand, it is clear from the discussion in section 7.1 that, in the absence of noise, the theoretical solution is a stationary point of the extended QZ-algorithm; in Figure 9.1 we see that the algorithm was still reliable in the presence

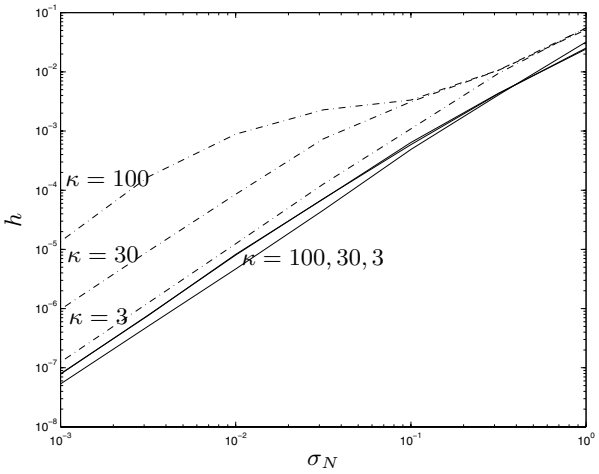


FIG. 9.2. *The mean value of h, defined in* (5.6)*, as a function of the noise level σ_N, for the Jacobi-algorithm (solid) and the extended QZ-algorithm (dashdot). The upper, middle, and lower curves correspond to a condition number κ of $\mathbf{U}^{(1)}$, equal to* 100*,* 30*, and* 3*, respectively.*

of noise.

In a second series of experiments we illustrate the convergence behavior of methods (a) and (c). For each of the sets of singular values of $\mathbf{U}^{(1)}$ (3, 2, 1; 30, 15, 1; 100, 50, 1), 100 independent samples of $\mathcal{A}$ are realized as before, with $\sigma_N = 0$. The different algorithms are now terminated if the instantaneous value of h has been reduced below $1e - 14$.

In Figure 9.3 we have plotted the average evolution of the value of h as a function of the iteration step l (for the scenarios with condition number $\kappa = 3, 30, 100$) and for algorithm (a) (only with condition number $\kappa = 3$, as will be motivated immediately). For these curves, the convergence speed is quasi-linear. The curves for the extended QZ-iteration have only been marginally affected by the chosen value of κ. On the other hand, it makes less sense to plot an average curve for the Jacobi-method in the cases where $\kappa = 30$ or 100, as the results can be strongly data-dependent. Namely, the convergence is still good in most cases, but for some particular instances of $\mathcal{A}$, the algorithm is observed to move through a swamp: apparently, like ALS iterations, Jacobi-iterations can be affected by swamps, although for well-conditioned problems they seem to form a minor issue. The extended QZ-algorithm appears to be less vulnerable, as the typical swamp behavior has only been observed for one instance of $\mathcal{A}$ ($\kappa = 100$). Rather than plotting the remaining mean convergence curves, we show in a histogram how many iterations were needed to terminate the algorithm in the different Monte Carlo simulations, for the various set-ups (see Figure 9.4, in which we have taken as a convention that experiments in which more than 100 iterations were needed, are added to the final histogram bin). Concerning Figures 9.3 and 9.4, we finally remark that for 5 instances of $\mathcal{A}$, the extended QZ-algorithm was observed to start by increasing the value of h to some extent, before actual convergence.

With respect to Figure 9.4 we conclude that for good condition numbers, the Jacobi-algorithm requires less iterations than the extended QZ-iteration. However, when the condition number increases, the risk increases that the Jacobi-algorithm requires a higher number of iterations. In this respect, we should also keep in mind

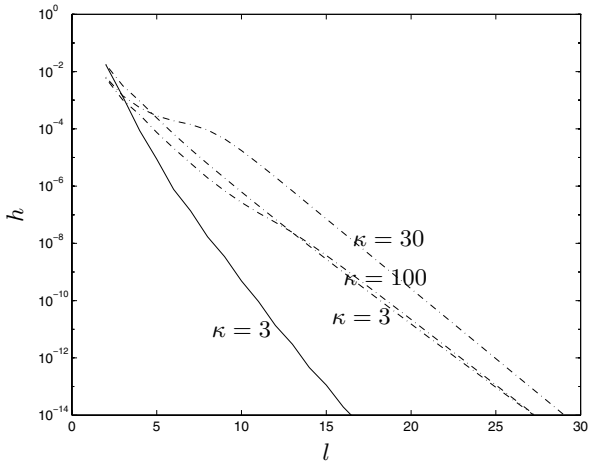


FIG. 9.3. *The evolution of h, defined in* (5.6)*, as a function of the iteration step l, for the Jacobi-algorithm (solid) and the extended QZ-algorithm (dashdot). The parameter κ is the condition number of $\mathbf{U}^{(1)}$.*

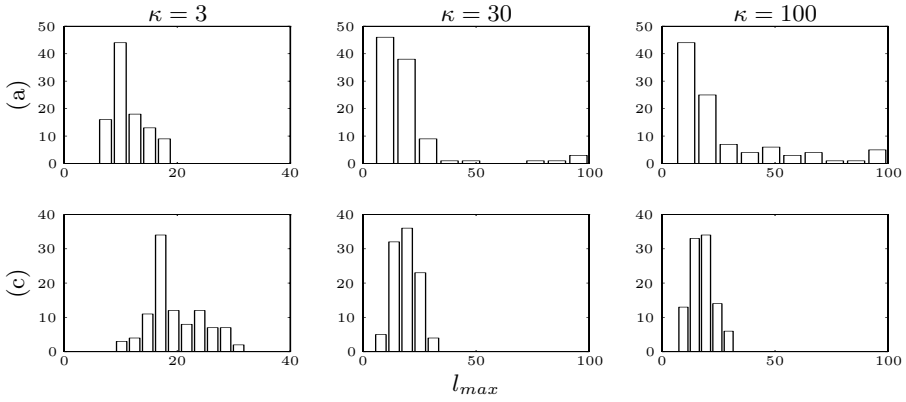


FIG. 9.4. *Histogram of* 100 *Monte Carlo simulations, showing the number of iterations required to reduce the value of h (see* (5.6)*) below* 1e−14*, for methods* (a) *(Jacobi-algorithm) and* (c) *(extended QZ-algorithm), and a condition number κ of $\mathbf{U}^{(1)}$, equal to* 3*,* 30*, or* 100*. Experiments in which more than* 100 *iterations were needed have been added to the final histogram bin.*

that for small tensors the computational complexity of a Jacobi-iteration step is higher than that of an extended QZ-iteration step; for larger tensor sizes, the extended QZ-iteration steps are more complex (as is clear from the discussion in section 7).

Figure 9.5 is an example of an ALS iteration moving through a "swamp." After 5 iterations the convergence speed becomes almost equal to zero, and after 70 iterations it starts to increase again. It is clear that tolerances have to be set very tight in order to reach the global optimum. This type of convergence is not uncommon. The figure was obtained for a $(2 \times 2 \times 4)$ tensor of the form (9.1), with the condition numbers of $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, and $\mathbf{U}^{(3)}$ equal to 2 and $\sigma_N = 1e-2$. Note that in this case (7.15) provides an explicit expression for the solution.

In the following experiment we will compare the performance of the simultaneous generalized Schur approach with other techniques. In each of 50 Monte Carlo runs, a tensor $\mathcal{A} \in \mathbb{R}^{2\times 2\times 10}$ of the form (9.1) is generated. The singular values of $\mathbf{U}^{(2)}$ are taken equal to $2, 1$. For $\mathbf{U}^{(1)}$ three different sets of singular values are considered:

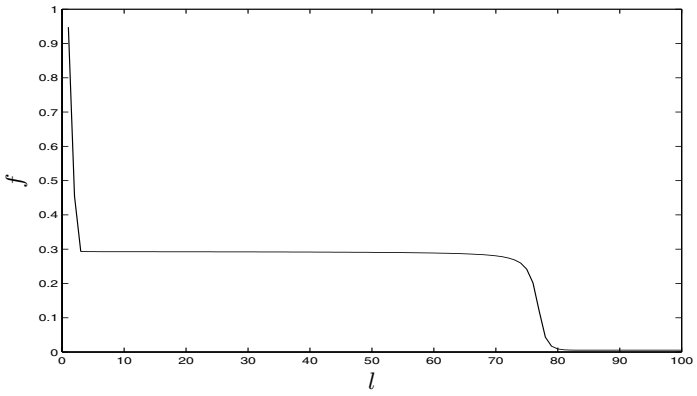


FIG. 9.5. *Example of a "swamp"-type convergence curve for ALS iterations. f is the cost function defined in* (2.3) *and l the iteration step.*

2,1; 10,1; 100,1. The entries of $\mathbf{U}^{(3)}$ are generated as $u_{ij}^{(3)} = 1 + g_{ij}/50$, in which g_{ij} is drawn from a Gaussian distribution with unit variance. For each particular choice of $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, $\mathbf{U}^{(3)}$, and $\tilde{\mathcal{N}}$, the scalar σ_N is varied between $1e-4$ and $1e-2$. In this way, σ_N ranges from a level where the eigenvalues in (4.10) are subject only to a small perturbation to a level where there is a certain risk that these eigenvalues have crossed each other.

In Figure 9.6 we compare the mean value of $\|\hat{\mathbf{U}}^{(1)} - \mathbf{U}^{(1)}\|$ obtained with a SGSD to the one obtained from the EVD of the matrix $\mathbf{V}_2 \cdot \mathbf{V}_1^{-1}$ (cf. [38, 43, 5, 42]). It is clear that the SGSD is more accurate than a single EVD, because it takes all the matrices $\mathbf{V}_k$ into account. However, the technique is more sensitive to the condition number of $\mathbf{U}^{(1)}$. In the case of an ill-conditioned matrix $\mathbf{U}^{(1)}$, the performance may considerably degrade when the noise level is high; as such, this effect cannot be examined by means of the first-order perturbation analysis in section 6. Note that the EVD may yield complex eigenvalues and eigenvectors for low signal-to-noise ratios.

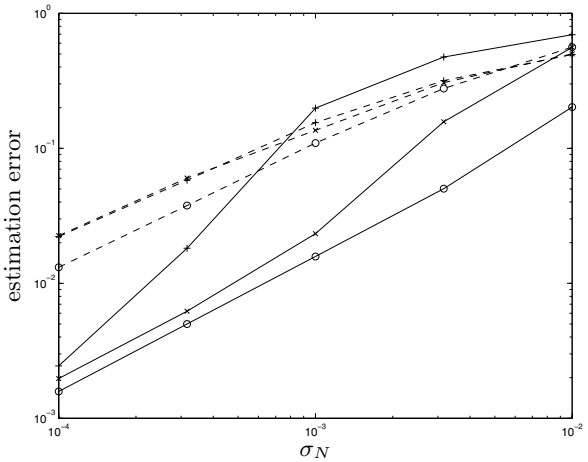


FIG. 9.6. *The mean value of $\|\hat{\mathbf{U}}^{(1)} - \mathbf{U}^{(1)}\|$, as a function of the noise level σ_N, for the SGSD (solid) and for a single EVD (dashed). The condition number κ of $\mathbf{U}^{(1)}$ is equal to* 2 (∘), 10 (×), *or* 100 (+).

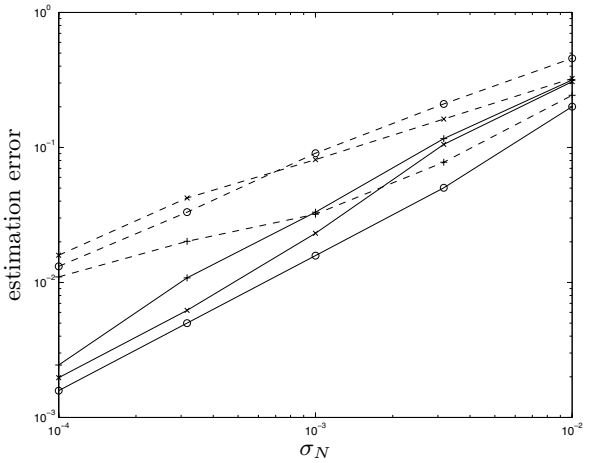


FIG. 9.7. *The mean value of* $\|\hat{\mathbf{U}}^{(1)} - \mathbf{U}^{(1)}\|$, *as a function of the noise level* σ_N, *for the SGSD, followed by an ALS iteration (solid) and for an EVD followed by an ALS iteration (dashed). The condition number* κ *of* $\mathbf{U}^{(1)}$ *is equal to* 2 *(*$\circ$*),* 10 *(*$\times$*), or* 100 *(+).*

In Figure 9.7 we display the accuracy obtained when the results of Figure 9.6 are used to initialize an ALS routine. The iteration was terminated when

$$\left\| \begin{pmatrix} \hat{\mathbf{U}}_{k+1}^{(1)} \\ \hat{\mathbf{U}}_{k+1}^{(2)} \\ \hat{\mathbf{U}}_{k+1}^{(3)} \end{pmatrix} - \begin{pmatrix} \hat{\mathbf{U}}_{k}^{(1)} \\ \hat{\mathbf{U}}_{k}^{(2)} \\ \hat{\mathbf{U}}_{k}^{(3)} \end{pmatrix} \right\| < 1e-4,$$

in which $\hat{\mathbf{U}}_k^{(i)}$ is the estimate of $\mathbf{U}^{(i)}$ at iteration step k. We see that, even after an ALS iteration, the EVD approach remains less accurate than the simultaneous generalized Schur approach. In additional simulations we observed that this is less the case when $\mathbf{U}^{(3)}$ is better conditioned.

In Figure 9.8 we put the result obtained by the SGSD and the enhanced result obtained by an extra ALS iteration next to each other. It turns out that the performance degradation that is linked to a bad condition number of $\mathbf{U}^{(1)}$ (as mentioned in the discussion of Figure 9.6), can be mitigated by an additional ALS iteration. If there is no such problem, then an extra ALS iteration is not required.

In Figure 9.9 we compare the ALS-enhanced SGSD result to the best result obtained by ALS iteration, starting from 10 random initializations. Remarkably enough, ALS gives better results when the condition number of $\mathbf{U}^{(1)}$ increases. The SGSD turns out to be more accurate than the direct ALS approach. In additional simulations we observed that the difference in performance decreases when $\mathbf{U}^{(3)}$ is better conditioned.

In Figure 9.10 we plot the total CPU time, over 50 Monte Carlo runs and 10 random initializations per run, required by the ALS routine. Analysis of the data showed that, for a given value of κ and σ_N, not more than 2 out of 3 initializations led to an estimation error $\|\hat{\mathbf{U}}^{(1)} - \mathbf{U}^{(1)}\|$ that was more than twice its minimal value over all runs and initializations. This ratio depended little on the particular value of κ and σ_N. This means that we actually did not have to start from 10 random initializations; 5 initializations would have been sufficient and the computational cost can be divided by two. Nevertheless, this remains much more expensive than a SGSD

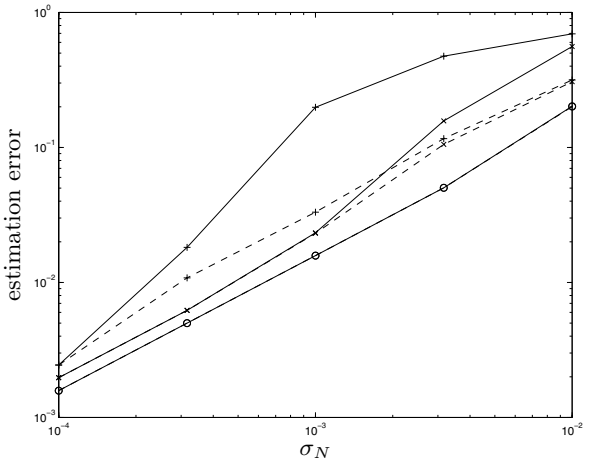


FIG. 9.8. *The mean value of* $\|\hat{\mathbf{U}}^{(1)} - \mathbf{U}^{(1)}\|$, *as a function of the noise level* σ_N, *obtained by the SGSD (solid), and by a subsequent ALS iteration (dashed). The condition number* κ *of* $\mathbf{U}^{(1)}$ *is equal to* 2 *(*$\circ$*),* 10 *(*$\times$*), or* 100 *(+).*

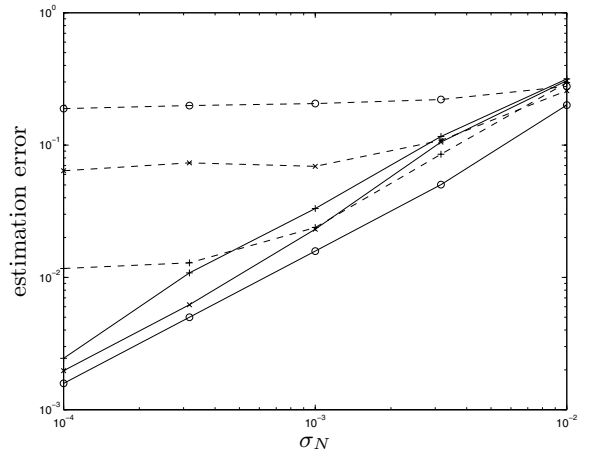


FIG. 9.9. *The mean value of* $\|\hat{\mathbf{U}}^{(1)} - \mathbf{U}^{(1)}\|$, *as a function of the noise level* σ_N, *for the ALS-enhanced SGSD (solid) and direct ALS starting from* 10 *random initializations (dashed). The condition number* κ *of* $\mathbf{U}^{(1)}$ *is equal to* 2 *(*$\circ$*),* 10 *(*$\times$*), or* 100 *(+).*

(overall CPU time approximately 1 s, independent of κ and σ_N) or a simple EVD (CPU time in the order of magnitude of $1e-2$ s) (the latter merely consists of MATLAB's function `eig` applied to a (2×2) matrix).

Figure 9.11 shows the CPU time required by the ALS iteration that was initialized by means of the SGSD or the EVD. Fewer computations were needed for the SGSD. The figure also shows that each of these two special initializations led to fewer ALS iterations than an average random start.

Whenever in this section we have used ALS iterations for the optimization of cost function f, we have also tried the general-purpose Levenberg–Marquardt algorithm [39] (we used the command `lsqnonlin` of the Optimization Toolbox 2.0 of MATLAB 5.3). In the last series of experiments, Levenberg–Marquardt gave consistently much less accurate results than ALS, even when the tolerance on the value of f was set as

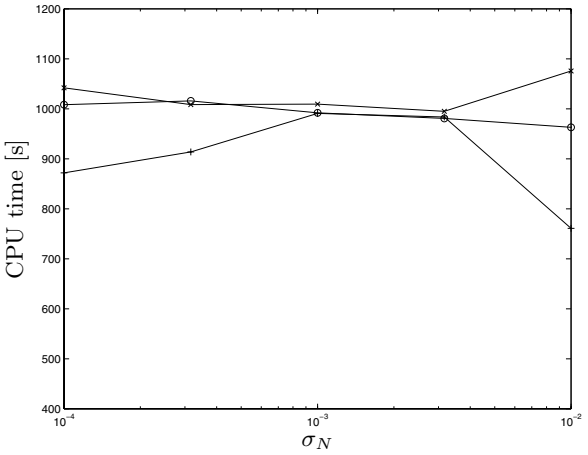


FIG. 9.10. *Total CPU time, over* 50 *Monte Carlo runs and* 10 *random initializations per run, required by the ALS routine. The condition number* κ *of* $\mathbf{U}^{(1)}$ *is equal to* 2 *(*$\circ$*),* 10 *(*$\times$*), or* 100 *(+).*

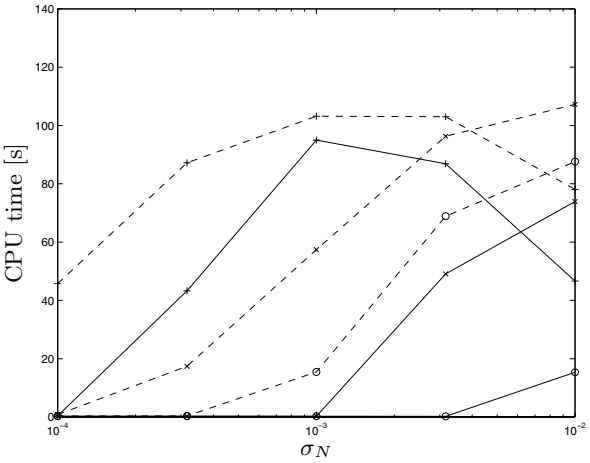


FIG. 9.11. *Total CPU time, over* 50 *Monte Carlo runs, required by the ALS iteration following a SGSD (solid) or a single EVD (dashed). The condition number* κ *of* $\mathbf{U}^{(1)}$ *is equal to* 2 *(*$\circ$*),* 10 *(*$\times$*), or* 100 *(+).*

sharp as $1e{-}10$. For well-conditioned problems, the accuracy of Levenberg–Marquardt and ALS may be comparable. However, in this case, Levenberg–Marquardt is typically an order of magnitude more expensive than ALS.

Finally, we have applied Algorithm 1 to a real-life dataset. It concerns a real-valued $(5 \times 10 \times 13)$-tensor representing the displacements of 13 points of the tongue of 5 test persons while pronouncing 10 vowels. A detailed description of these data and their analysis by means of a CANDECOMP can be found in [29]. The dataset can be downloaded from [21].

First, we observed that the two dominant 1-mode, 2-mode, and 3-mode singular values [19] explain 94.5%, 95.4%, and 96.0%, respectively, of the "energy" in the dataset. Therefore we performed a dimensionality reduction by calculating the best rank-$(2, 2, 2)$ approximation of the data tensor before starting the actual CANDECOMP computations, as explained in section 3. The approximation was obtained by

means of a higher-order orthogonal iteration, initialized with the truncated HOSVD [20]. The stop criterion consisted of checking if the adjustment of each of the component matrices in an iteration step was below $1e-4$ (Frobenius-norm). The algorithm converged in 5 steps. The approximation contained 92.6% of the energy.

Next, we looked for the least-squares approximation of the $(2\times2\times2)$-core tensor by a sum of two rank-1 components. We resorted to the Jacobi-technique of section 7.2, in which the solution was found by rooting a polynomial of degree 8. The error of the fit was in the order of the numerical accuracy of MATLAB. Backtransformation to the original dimensionality by multiplication with the best rank-$(2,2,2)$ components yielded the best rank-2 approximation of the original dataset. Further enhancement by an additional minimization of cost function (2.3) was not possible (step 5 in Algorithm 1); otherwise, the rank-$(2,2,2)$ approximation would not have been optimal or the core tensor would not have been rank-2.

The cosine of the angle in $\mathbb{R}^{5\times10\times13}$ between the original data tensor and its rank-2 approximation was equal to 0.962, which was even slightly better than the result of [29] (0.956); the latter result had been obtained by repeating ALS iterations for different rank estimates and different starting values, and cross-examining the results. On a SUN Ultra 2 Sparc and using MATLAB 4.2c, our computations took $0.2+0.04s$ of CPU-time, which was a drastic improvement [37].

10. Conclusion. In this paper we have investigated the computation of the CANDECOMP, under the assumptions made in section 2. Currently, the calculation of the factors mostly takes the form of an ALS descent algorithm, possibly initialized with an estimate obtained by a matrix EVD. For well-conditioned problems ALS iterations are reliable. However, for some ill-conditioned problems the results are less satisfactory. In this paper the CANDECOMP is computed via a simultaneous diagonalization, by equivalence or congruence, of a set of matrices. Since we take all the available information into account, this is numerically more reliable than the calculation of a single EVD. Diagonalization by a simultaneous congruence transformation was encountered as well in the derivation of an analytical constant modulus algorithm [48], where it was translated into a SGSD and subsequently solved by means of an extended QZ-iteration scheme. In this paper, we have also proposed a Jacobi-type algorithm. In this context we have derived the explicit solution for the case of rank-2 tensors. The behavior of the different algorithms was illustrated and their performance compared by means of some numerical experiments. In this paper we have also studied necessary and sufficient conditions for the uniqueness of some simultaneous matrix decompositions; in addition, we have performed a first-order perturbation analysis of the SGSD.

Acknowledgment. The authors wish to thank Dr. J. Dehaene (K.U.Leuven) for explaining the basic principles underlying the derivation in section 6.2.

REFERENCES

[1] C.J. Appellof and E.R. Davidson, *Strategies for analyzing data from video fluoromatric monitoring of liquid chromatographic effluents*, Analytical Chemistry, 53 (1981), pp. 2053–2056.

[2] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, *A blind source separation technique using second-order statistics*, IEEE Trans. Signal Process., 45 (1997), pp. 434–444.

[3] R. Bro, *PARAFAC. Tutorial & applications*, Chemom. Intell. Lab. Syst., 38 (1997), pp. 149–171.

[4] A. Bunse-Gerstner, R. Byers, and V. Mehrmann, *Numerical methods for simultaneous diagonalization*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 927–949.

[5] D. Burdick, X. Tu, L. McGown, and D. Millican, *Resolution of multicomponent fluorescent mixtures by analysis of the excitation-emission-frequency array*, J. Chemometrics, 4 (1990), pp. 15–28.

[6] J.-F. Cardoso, *Super-symmetric decomposition of the fourth-order cumulant tensor. Blind identification of more sources than sensors*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Toronto, Canada, 1991, pp. 3109–3112.

[7] J.-F. Cardoso, *Iterative techniques for blind source separation using only fourth-order cumulants*, in Signal Processing VI: Theories and Applications, Proc. EUSIPCO-92, Brussels, Belgium, 1992, pp. 739–742.

[8] J.-F. Cardoso and A. Souloumiac, *Blind beamforming for non-Gaussian signals*, IEE Proc.-F, 140 (1994), pp. 362–370.

[9] J.-F. Cardoso and A. Souloumiac, *Jacobi angles for simultaneous diagonalization*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 161–164.

[10] J. Carroll and J. Chang, *Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition*, Psychometrika, 9 (1970), pp. 267–283.

[11] J. Carroll, G. Desoete, and S. Pruzansky, *An evaluation of five algorithms for generating an initial configuration for SINDSCAL*, J. Classification, 6 (1989), pp. 105–119.

[12] M.T. Chu, *A continuous Jacobi-like approach to the simultaneous reduction of real matrices*, Linear Algebra Appl., 147 (1991), pp. 75–96.

[13] P. Comon, *Independent component analysis, a new concept?* Signal Process., 36 (1994), pp. 287–314.

[14] P. Comon and B. Mourrain, *Decomposition of quantics in sums of powers of linear forms*, Signal Process., 53 (1996), pp. 93–108.

[15] J. Dehaene, *Continuous-Time Matrix Algorithms, Systolic Algorithms and Adaptive Neural Networks*, Ph.D. Thesis, E.E. Dept. (ESAT), K.U.Leuven, Belgium, 1995.

[16] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Independent component analysis based on higher-order statistics only*, in Proceedings of the IEEE Signal Processing Workshop on Statistical Signal and Array Processing, Corfu, Greece, 1996, pp. 356–359.

[17] L. De Lathauwer, *Signal Processing Based on Multilinear Algebra*, Ph.D. thesis, K.U. Leuven, E.E. Dept. (ESAT), Belgium, 1997.

[18] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Jacobi-algorithm for simultaneous generalized Schur decomposition in higher-order-only ICA*, in Proceedings of the IEEE Benelux Signal Processing Chapter Signal Processing Symposium (SPS'98), Leuven, Belgium, 1998, pp. 67–70.

[19] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.

[20] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.

[21] B. De Moor, Ed., *Database for the Identification of Systems (DAISY)*, E.E. Dept., ESAT/SCD, K.U.Leuven, Belgium, http://www.esat.kuleuven.ac.be/sista/daisy/.

[22] A. Edelman, T.A. Arias, and S.T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.

[23] B. Flury, *Common Principal Components & Related Multivariate Models*, John Wiley & Sons, New York, 1988.

[24] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[25] M. Haardt, K. Hüper, J. Moore, and J. Nossek, *Simultaneous Schur Decomposition of several matrices to achieve automatic pairing in multidimensional harmonic retrieval problems*, in Signal Processing VIII: Theories and Applications, Proceedings of EUSIPCO-96, Trieste, Italy, 1996, Vol. 1, pp. 531–534.

[26] R. Harshman, *Foundations of the PARAFAC procedure*: *Model and conditions for an "explanatory" multi-mode factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[27] R.A. Harshman and M.E. Lundy, *The PARAFAC model for three-way factor analysis and multidimensional scaling*, in Research Methods for Multimode Data Analysis, H.G. Law, C.W. Snyder, J.A. Hattie, and R.P. McDonald, eds., Praeger, NY, 1984, pp. 122–215.

[28] R.A. Harshman and M.E. Lundy, *PARAFAC*: *Parallel factor analysis*, Comput. Statist. Data

Anal., 18 (1994), pp. 39–72.
[29] R. Harshman, P. Ladefoged, and L. Goldstein, *Factor analysis of tongue shapes*, J. Acoust. Soc. Am., 62 (1977), pp. 693–707.
[30] T. Kato, *A Short Introduction to Perturbation Theory for Linear Operators*, Springer-Verlag, New York, 1982.
[31] H. Kiers and W. Krijnen, *An efficient algorithm for PARAFAC of three-way data with large numbers of observational units*, Psychometrika, 56 (1991), pp. 147–152.
[32] E. Kofidis and P.A. Regalia, *On the best rank-1 approximation of higher-order supersymmetric tensors*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 863–884.
[33] T. 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.
[34] J.B. Kruskal, *Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear Algebra Appl., 18 (1977), pp. 95–138.
[35] J.B. Kruskal, *Rank, decomposition, and uniqueness for 3-way and N-way arrays*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., North-Holland, Amsterdam, 1989, pp. 7–18.
[36] J.B. Kruskal, R.A. Harshman, and M.E. Lundy, *How 3-MFA data can cause degenerate PARAFAC solutions, among other relationships*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., North-Holland, Amsterdam, 1989, pp. 115–122.
[37] P. Ladefoged, *personal communication*, Linguistics Dept., UCLA, Los Angeles, CA, 1996.
[38] S.E. Leurgans, R.T. Ross, and R.B. Abel, *A decomposition for three-way arrays*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1064–1083.
[39] J.J. More, *The Levenberg-Marquardt algorithm: implementation and theory*, in Numerical Analysis, Lecture Notes in Math. 630, G.A. Watson, ed., Springer-Verlag, Berlin, 1977, pp. 105–116.
[40] P. Paatero, *A weighted non-negative least squares algorithm for three-way "PARAFAC" factor analysis*, Chemom. Intell. Lab. Syst., 38 (1997), pp. 223–242.
[41] W.S. Rayens and B.C. Mitchell, *Two-factor degeneracies and a stabilization of PARAFAC*, Chemom. Intell. Lab. Syst., 38 (1997), pp. 173–181.
[42] E. Sanchez and B.R. Kowalski, *Tensorial resolution: A direct trilinear decomposition*, J. Chemometrics, 4 (1990), pp. 29–45.
[43] R. Sands and F. Young, *Component models for three-way data: An alternating least squares algorithm with optimal scaling features*, Psychometrika, 45 (1980), pp. 39–67.
[44] R. Schmidt, *A Signal Subspace Approach to Multiple Emitter Location and Spectral Estimation*, Ph.D. thesis, Stanford University, 1981.
[45] N. Sidiropoulos, G. Giannakis, and R. Bro, *Blind PARAFAC receivers for DS-CDMA systems*, IEEE Trans. Signal Process., 48 (2000), pp. 810–823.
[46] N. Sidiropoulos, R. Bro, and G. Giannakis, *Parallel factor analysis in sensor array processing*, IEEE Trans. Signal Process., 48 (2000), pp. 2377–2388.
[47] N. Sidiropoulos and R. Bro, *On the uniqueness of multilinear decomposition of N-way arrays*, J. Chemometrics, 14 (2000), pp. 229–239.
[48] A.-J. van der Veen and A. Paulraj, *An analytical constant modulus algorithm*, IEEE Trans. Signal Process., 44 (1996), pp. 1136–1155.
[49] A. Yeredor, *Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation*, IEEE Trans. Signal Process., 50 (2002), pp. 1545–1553.
[50] T. Zhang and G.H. Golub, *Rank-one approximation to high order tensors*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 534–550.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 2, pp. 328–349

MODEL REDUCTION OF MIMO SYSTEMS VIA TANGENTIAL INTERPOLATION*

K. GALLIVAN†, A. VANDENDORPE‡, AND P. VAN DOOREN‡

Abstract. In this paper, we address the problem of constructing a reduced order system of minimal McMillan degree that satisfies a set of tangential interpolation conditions with respect to the original system under some mild conditions. The resulting reduced order transfer function appears to be generically unique and we present a simple and efficient technique to construct this interpolating reduced order system. This is a generalization of the *multipoint Padé* technique which is particularly suited to handle multiinput multioutput systems.

Key words. linear time invariant, multivariable system, model reduction, tangential interpolation, Krylov method, multipoint Padé

AMS subject classifications. 41A21, 65F99

DOI. 10.1137/S0895479803423925

1. Introduction.

Model reduction of large-scale dynamical systems has received a lot of attention during the last decade: it is a crucial tool in reducing the computational complexity of, e.g., analysis and design of micro-electro-mechanical systems (MEMS) [13], in simulation of electronic devices [5], in weather prediction [6], and in control of partial differential equations [12].

The construction of the reduced order model typically passes via the derivation of one or two projective subspaces of the state space in which the original system is modelled. There are several approaches to find such projective subspaces. In this paper, we focus on an approach related to tangential interpolation of the rational transfer function, which therefore only works for linear time invariant systems. Tangential interpolation of given input/output data has already been treated in the literature [3], [4]. Here, we address the case where these data are themselves obtained from tangential information of a given (large-scale) transfer function, which to our knowledge has not been considered.

In this paper, we consider $p \times m$ *strictly proper* transfer functions $T(s)$, i.e., where $\lim_{s\to\infty} T(s) = 0$. This implies that the point at infinity is a zero of $T(s)$. For this reason, a separate treatment of the point at infinity is required.

We begin with some definitions which will allow us to formalize the problem of tangential interpolation. We say that a rational matrix function $R(s)$ is $O(\lambda - s)^k$ in s with $k \in \mathbb{Z}$ if its Taylor expansion about the point λ can be written as follows:

*Received by the editors March 11, 2003; accepted for publication (in revised form) September 18, 2003; published electronically November 17, 2004. This paper presents research supported by the Belgian Programme on Inter-university Poles of Attraction and initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture. This work was also supported by the National Science Foundation under grant CCR-9912415.

http://www.siam.org/journals/simax/26-2/42392.html

†School of Computational Science and Information Technology, Florida State University, Tallahassee, FL 32306-4120 (gallivan@csit.fsu.edu).

‡Université Catholique de Louvain, Department of Mathematical Engineering, Batiment Euler (A. 119), 4 avenue Georges Le maitre, B-1348 Louvain-la-Neuve, Belgium (vandendorpe@csam.ucl.ac.be, vdooren@csam.ucl.ac.be). The work of this author was supported by a research fellowship from the Belgian National Fund for Scientific Research.

$$R(s) = O(\lambda - s)^k \Longleftrightarrow R(s) = \sum_{i=k}^{+\infty} R_i(\lambda - s)^i, \tag{1.1}$$

where the coefficients R_i are constant matrices. If $R_k \neq 0$, then we say that $R(s) = \Theta(\lambda - s)^k$. As a consequence, if $R(s) = \Theta(\lambda - s)^k$ and k is strictly negative, then λ is a pole of $R(s)$, and if k is strictly positive, then λ is a zero of $R(s)$. Analogously, we say that $R(s)$ is $O(s^{-1})^k$ if the following condition is satisfied:

$$R(s) = O(s^{-1})^k \Longleftrightarrow R(s) = \sum_{i=k}^{+\infty} R_i s^{-i}, \tag{1.2}$$

where the coefficients R_i are constant matrices. It should be stressed that, in general, $R(s)$ being $O(s)^{-k}$ is not equivalent to $R(s)$ being $O(s^{-1})^k$.

We must also use the well-established concept of a zero of a system (see, e.g., [14]) and the following related definition.

DEFINITION 1.1. *Suppose that $T(s)$ is a $p \times m$ rational function. The zeros of the numerator polynomials not equal to zero in the Smith–McMillan form of the transfer function $T(s)$ are called the zeros of $T(s)$. An $m \times 1$ polynomial vector $y(s)$ is a right zero direction of order k at λ if $y(\lambda) \neq 0$ and*

$$T(s)y(s) = O(\lambda - s)^k. \tag{1.3}$$

Analogously, a $1 \times p$ polynomial vector $x(s)$ is a left zero direction of $T(s)$ when $x^*(s)$ is a right zero of $T^*(s)$. The order of a zero is defined as the maximum order of the zero directions at this point.

For MIMO systems, a zero can also be a pole. If λ is not a pole of $T(s)$, only the k first Taylor coefficients of $y(s)$ about λ are important. If λ is a pole of $T(s)$, the situation is more complicated. Indeed, assume that λ is a pole of order p of $T(s)$ and that $y(s)$ has an expansion about λ; then

$$T(s)y(s) = \left(\sum_{i=-p}^{+\infty} T_i(\lambda - s)^i \right) \left(\sum_{j=0}^{\infty} y_j(\lambda - s)^j \right). \tag{1.4}$$

We see that the first $k + p$ terms in the Taylor expansion of $y(s)$ are important to ensure that the product (1.4) has a zero of order k. This case will not be discussed in this paper, but a few remarks will be made to indicate how it complicates the problem.

We now present the concept of tangential interpolation that will be considered in this paper. Three concepts are defined, namely left, right, and two-sided tangential interpolation. Interpolation at the point at infinity is considered as a special case.

Let z be a finite point in the complex plane. Let $T(s)$ and $\hat{T}(s)$ be two $p \times m$ strictly proper transfer functions that do not have a pole at $s = z$.

Left tangential interpolation. Let $x(s)$ be a $1 \times p$ polynomial vector of degree $\beta - 1$ and not equal to zero at $s = z$. We say that $\hat{T}(s)$ interpolates $T(s)$ at $(z, x(s))$ if

$$x(s)(T(s) - \hat{T}(s)) = O(z - s)^\beta. \tag{1.5}$$

Let $x(s)$ be a $1 \times p$ polynomial vector in s^{-1}, of degree $\beta - 1$ in s^{-1} and not equal to zero at $s = \infty$. We say that $\hat{T}(s)$ interpolates $T(s)$ at $(\infty, x(s))$ if

$$x(s)(T(s) - \hat{T}(s)) = O(s^{-1})^{\beta+1}. \tag{1.6}$$

Right tangential interpolation. Let $y(s)$ be a $m \times 1$ polynomial vector of degree $\delta - 1$ and not equal to zero at $s = z$. We say that $\hat{T}(s)$ interpolates $T(s)$ at $(z, y(s))$ if

$$(T(s) - \hat{T}(s))y(s) = O(z - s)^{\delta}. \tag{1.7}$$

Let $y(s)$ be a $m \times 1$ polynomial vector in s^{-1}, of degree $\delta - 1$ in s^{-1} and not equal to zero at $s = \infty$. We say that $\hat{T}(s)$ interpolates $T(s)$ at $(\infty, y(s))$ if the following condition is satisfied:

$$(T(s) - \hat{T}(s))y(s) = O(s^{-1})^{\delta+1}. \tag{1.8}$$

Two-sided tangential interpolation. Let $x(s)$ be a $1 \times p$ polynomial vector of degree $\beta - 1$ and not equal to zero at $s = z$. Let $y(s)$ be a $m \times 1$ polynomial vector of degree $\delta - 1$ and not equal to zero at $s = z$. We say that $\hat{T}(s)$ interpolates $T(s)$ at $(z, x(s), y(s))$ if the following condition is satisfied:

$$x(s)(T(s) - \hat{T}(s))y(s) = O(z - s)^{\beta+\delta}. \tag{1.9}$$

Let $x(s)$ be a $1 \times p$ polynomial vector in s^{-1}, of degree $\beta - 1$ in s^{-1} and not equal to zero at $s = \infty$. Let $y(s)$ be a $m \times 1$ polynomial vector in s^{-1}, of degree $\delta - 1$ in s^{-1} and not equal to zero at $s^{-1} = 0$. We say that $\hat{T}(s)$ interpolates $T(s)$ at $(\infty, x(s), y(s))$ if the following condition is satisfied:

$$x(s)(T(s) - \hat{T}(s))y(s) = O(s^{-1})^{\beta+\delta+1}. \tag{1.10}$$

The objective of this paper is the following. We are given a transfer function $T(s)$ and a set of tangential interpolation conditions of the type (1.5) to (1.10) in a number of points of the complex plane, and we want to construct the transfer function of minimal McMillan degree that satisfies these interpolation conditions. In order to make the problem more precise, we need to introduce the following concepts.

DEFINITION 1.2. *Let $z_1, \ldots, z_{k_{left}}$ be points in the complex plane, not necessarily distinct or finite. For each finite z_α, a $1 \times p$ polynomial vector $x_\alpha(s)$ of degree $\beta_\alpha - 1$ and not equal to zero at $s = z_\alpha$ is given:*

$$x_\alpha(s) = \sum_{j=0}^{\beta_\alpha - 1} x_\alpha^{[j]}(z_\alpha - s)^j, \quad x_\alpha^{[0]} \neq 0. \tag{1.11}$$

If $z_\alpha = \infty$, then a $1 \times p$ polynomial vector in s^{-1}, $x_\alpha(s)$ of degree $\beta_\alpha - 1$ in s^{-1} and not equal to zero at $s = \infty$ is given:

$$x_\alpha(s) = \sum_{j=0}^{\beta_\alpha - 1} x_\alpha^{[j]} s^{-j}, \quad x_\alpha^{[0]} \neq 0. \tag{1.12}$$

The left interpolation set I_{left} is defined as follows:

$$I_{left} \doteq \left\{ (z_1, x_1(s)), \ldots, \left(z_{k_{left}}, x_{k_{left}}(s)\right) \right\}. \tag{1.13}$$

The size of I_{left}, written $s(I_{left})$, is defined as follows:

$$s(I_{left}) \doteq \sum_{i=1}^{k_{left}} \beta_i. \tag{1.14}$$

Finally, the set of interpolation points of I_{left}, written $p(I_{left})$ is defined as follows:

$$p(I_{left}) = \{z_1, \ldots, z_{k_{left}}\}. \tag{1.15}$$

Analogously, a right tangential interpolation set

$$I_{right} \doteq \left\{(w_1, y_1(s)), \ldots, \left(w_{k_{right}}, y_{k_{right}}(s)\right)\right\}, \tag{1.16}$$

with the points $w_1, \ldots, w_{k_{right}}$ arbitrarily chosen in $\mathbb{C} \cup \infty$ and each $m \times 1$ polynomial vector $y_\alpha(s)$, $1 \le \alpha \le k_{right}$ of degree $\delta_\alpha - 1$ in s if w_α is finite (of degree $\delta_\alpha - 1$ in s^{-1} otherwise) defined with the same conventions as above.

Let I_l be a left tangential interpolation set. Let I_r be a right tangential interpolation set. The set

$$I = \{I_l, I_r\} \tag{1.17}$$

is called a tangential interpolation set. The set of interpolation points of I, written $p(I)$, is defined by

$$p(I) \doteq p(I_l) \cup p(I_r). \tag{1.18}$$

Let $T(s)$ be a transfer function, then we say that the tangential interpolation set I is $T(s)$-admissible if $T(s)$ has m inputs and p outputs and no point belonging to $p(I)$ is a pole of $T(s)$, i.e., no interpolation point is a pole of $T(s)$.

Let the tangential interpolation set $I = \{I_l, I_r\}$ be defined as above. If some $z_\alpha \in I_l$ is equal to some $w_\gamma \in I_r$, say $\xi_{\alpha,\gamma} = z_\alpha = w_\gamma$, then define $x_\alpha^{(f)}(s)$ to be the polynomial vector of size $1 \times p$ of degree f obtained by keeping the first f terms in the Taylor expansion of $x_\alpha(s)$ about z_α, and analogously for $y_\gamma^{(g)}(s)$:

$$x_\alpha^{(f)}(s) \doteq \sum_{j=0}^{f-1} x_\alpha^{[j]}(z_\alpha - s)^j, \quad y_\gamma^{(g)}(s) \doteq \sum_{j=0}^{g-1} y_\gamma^{[j]}(w_\gamma - s)^j. \tag{1.19}$$

Use the same notation if z_α or w_γ is equal to ∞:

$$x_\alpha^{(f)}(s) \doteq \sum_{j=0}^{f-1} x_\alpha^{[j]} s^{-j}, \quad y_\gamma^{(g)}(s) \doteq \sum_{j=0}^{g-1} y_\gamma^{[j]} s^{-j}. \tag{1.20}$$

We are now able to define the tangential interpolation problem.

DEFINITION 1.3. *Let $T(s)$ and $\hat{T}(s)$ be two* strictly proper *$p \times m$ transfer functions. $\hat{T}(s)$ interpolates $T(s)$ at I if the three following conditions are satisfied:*

1. *$\hat{T}(s)$ interpolates $T(s)$ at any couple $(z_\alpha, x_\alpha(s))$ belonging to I_l,*
2. *$\hat{T}(s)$ interpolates $T(s)$ at any couple $(w_\gamma, y_\gamma(s))$ belonging to I_r,*
3. *Finally, for every $z_\alpha = w_\gamma \doteq \xi_{\alpha,\gamma}$, we impose in addition that for all $f = 1, \ldots, \beta_\alpha$; $g = 1, \ldots, \delta_\gamma$, $\hat{T}(s)$ interpolates $T(s)$* at $(\xi_{\alpha,\gamma}, x_\alpha^{(f)}(s), y_\gamma^{(g)}(s))$.

Two remarks are in order. In this paper, we consider only the simple case when the interpolation set I is $T(s)$-admissible and $\hat{T}(s)$-admissible. Second, the tangential interpolation problem has been studied in a slightly different form in the literature, e.g., in [4], and the reader is directed there for general results about the theory of interpolation of rational matrix functions. At first sight, one could think that our definition of the two-sided tangential interpolation problem is not the same as the

one treated in [4]. A lemma showing the equivalence between the two formulations is proved in the appendix.

The problem solved in this paper can be stated as follows.

PROBLEM 1.1. *We are given a strictly proper $p \times m$ transfer function $T(s)$ of McMillan degree N, and a corresponding minimal state space realization (C, A, B), such that*

$$T(s) = C(sI_N - A)^{-1}B,$$

with $C \in \mathbb{C}^{p\times N}$, $A \in \mathbb{C}^{N\times N}$, and $B \in \mathbb{C}^{N\times m}$. We are also given a $T(s)$-admissible tangential interpolation set I. We want to construct a $p \times m$ reduced order transfer function $\hat{T}(s)$ of minimal McMillan degree n,

$$\hat{T}(s) = \hat{C}(sI_n - \hat{A})^{-1}\hat{B}, \tag{1.21}$$

with $\hat{C} \in \mathbb{C}^{p\times n}$, $\hat{A} \in \mathbb{C}^{n\times n}$, $\hat{B} \in \mathbb{C}^{n\times m}$ such that I is $\hat{T}(s)$-admissible and $\hat{T}(s)$ tangentially interpolates $T(s)$ at I.

The remainder of this paper is organized as follows. In section 2, the tangential interpolation problem is solved for two simple sets of interpolation conditions. In section 3, the background necessary to solve the general problem, Problem 1.1, is introduced. In section 4, the multipoint Padé approximation is constructed and its main properties are analyzed. Concluding remarks are given in section 5.

2. Preliminary results. In this section, we present the solution of Problem 1.1 for two particular interpolation sets. The general results are given in sections 3 and 4.

2.1. One set of n distinct right interpolation conditions. The first simpler problem solved in this section is the following.

PROBLEM 2.1. *Let $T(s)$ be a $p \times m$ transfer function of McMillan degree N. Let $\{\lambda_1, \ldots, \lambda_n\}$ be n (where $n < N$) distinct finite points in the complex plane that are not poles of $T(s)$. Let $\{y_1, \ldots, y_n\}$ be n $m \times 1$ nonzero vectors. We want to construct a $p \times m$ transfer function $\hat{T}(s)$ of McMillan degree n such that for all $1 \le i \le n$,*

$$T(\lambda)y_i = \hat{T}(\lambda_i)y_i. \tag{2.1}$$

Let C, A, B be a minimal state space realization of the $p \times m$ transfer function $T(s)$. In order to solve the problem, we construct the $N \times n$ matrix $V \doteq \begin{bmatrix} v_1 \ldots v_n \end{bmatrix}$ that satisfies the following Sylvester equation:

$$A \begin{bmatrix} v_1 \ldots v_n \end{bmatrix} - \begin{bmatrix} v_1 \ldots v_n \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} + B \begin{bmatrix} y_1 \ldots y_n \end{bmatrix} = 0. \tag{2.2}$$

Assume that V has full column rank n. Construct $Z \in \mathbb{C}^{N\times n}$ such that

$$Z^T V = I_n.$$

Construct $\hat{C} \in \mathbb{C}^{p\times n}$, $\hat{A} \in \mathbb{C}^{n\times n}$, and $\hat{B} \in \mathbb{C}^{n\times m}$ as follows:

$$\hat{C} \doteq CV, \quad \hat{A} \doteq Z^T AV, \quad \hat{B} \doteq Z^T B.$$

To verify that the transfer function

$$\hat{T}(s) \doteq \hat{C}(sI_n - \hat{A})^{-1}\hat{B}$$

solves Problem 2.1, first note that for any $1 \le i \le k$ the columns of V can be computed as follows:

$$v_i = (\lambda_i I_N - A)^{-1} B y_i.$$

We will also use the following well-known result.

LEMMA 2.1. *Let $V \in \mathbb{C}^{N\times n}$. If the vector v belongs to the column span of the matrix V. Then, for any matrix $W \in \mathbb{C}^{N\times n}$ such that $W^T V = I_k$,*

$$v = VW^T v.$$

Proof. Because v belongs to the linear span of the columns of V, there exists a vector $\hat{v} \in \mathbb{C}^n$ such that $v = V\hat{v}$. For any W^T satisfying $W^T V = I_n$, we have $\hat{v} = W^T v$. This in turn implies that $v = VW^T v$. ∎

Defining W by

$$W^T \doteq \left(Z^T(\lambda_1 I_N - A)V\right)^{-1} Z^T(\lambda_1 I_N - A).$$

clearly yields $W^T V = I_n$ and applying the preceding lemma, we obtain the following equalities:

$$T(\lambda_1)y_1 = C(\lambda_1 I_N - A)^{-1} B y_1 \tag{2.3}$$

$$= CVW^T(\lambda_1 I_N - A)^{-1} B y_1 \tag{2.4}$$

$$= CV(\lambda_1 I_k - Z^T A V)^{-1} Z^T B y_1 \tag{2.5}$$

$$= \hat{T}(\lambda_1) y_1. \tag{2.6}$$

This proves that $\hat{T}(s)$ solves Problem 2.1.

REMARK 2.1.

1. *This reasoning is very similar to the technique used in the SISO case in* [7] *and* [11] *. These papers develop techniques to construct a SISO transfer function of McMillan degree n that satisfies a set of (scalar) interpolation conditions with respect to an original transfer function.*
2. *It should be pointed out that the transfer function $\hat{T}(s)$ of McMillan degree n that solves Problem* 2.1 *is not unique. This is due to the fact that there exist infinitely many matrices $Z \in \mathbb{C}^{N\times n}$ such that $Z^T V = I_n$, where V satisfies* (2.2) *and is generically unique. We will see in what follows that, by imposing n additional left interpolation conditions, one generically determines a unique reduced order transfer function $\hat{T}(s)$ of McMillan degree n.*

2.2. One unique two-sided interpolation condition. We next consider the case where the interpolation set consists of only one finite interpolation point $\alpha \in \mathbb{C}$, i.e., in terms of the parameters of Problem 1.1,

$$k_{left} = k_{right} = 1, \quad \beta_1 = \delta_1 = n, \quad z_1 = w_1 = \alpha. \tag{2.7}$$

Moreover, we assume that α is not a pole of $T(s)$. Deleting the subscripts not required due to the simpler conditions to clarify the notation allows the problem to be stated as follows.

PROBLEM 2.2. *Given $T(s) = C(sI_N - A)^{-1}B$, $\alpha \in \mathbb{C}$, $x(s) \doteq \sum_{i=0}^{n-1} x^{[i]}(\alpha - s)^i$ and $y(s) \doteq \sum_{i=0}^{n-1} y^{[i]}(\alpha - s)^i$, construct a reduced order transfer function $\hat{T}(s)$ of McMillan degree n such that*

$$x(s)T(s) = x(s)\hat{T}(s) + O(\alpha - s)^n, \tag{2.8}$$

$$T(s)y(s) = \hat{T}(s)y(s) + O(\alpha - s)^n, \tag{2.9}$$

and for all $f = 1, \ldots, n$, $g = 1, \ldots, n$,

$$x^{(f)}(s)(T(s) - \hat{T}(s))y^{(g)}(s) = O(\alpha - s)^{f+g}. \tag{2.10}$$

In order to solve the problem, we first rewrite (2.8)–(2.10) as matrix equations. Note that for any $\alpha \in \mathbb{C}$ is not a pole of $T(s)$, we can write

$$T(s) = C(sI_N - A)^{-1}B = C\left((s-\alpha)I_N + \alpha I - A\right)^{-1}B \tag{2.11}$$

$$= C(\alpha I_N - A)^{-1}\left(I - (\alpha - s)(\alpha I - A)^{-1}\right)^{-1}B \tag{2.12}$$

$$= \sum_{k=0}^{\infty} C(\alpha I - A)^{-k-1}B(\alpha - s)^k. \tag{2.13}$$

Let us consider the left interpolation conditions corresponding to equation (2.8). By imposing the n first coefficients of the Taylor expansion of the product $x(s)(T(s) - \hat{T}(s))$ to be zero, we find the following system of equations:

$$\begin{aligned} &x^{[0]}C(\alpha I - A)^{-1}B \\ &= x^{[0]}\hat{C}(\alpha I - \hat{A})^{-1}\hat{B} \end{aligned} \tag{2.14}$$

$$\begin{aligned} &x^{[1]}C(\alpha I - A)^{-1}B + x^{[0]}C(\alpha I - A)^{-2}B \\ &= x^{[1]}\hat{C}(\alpha I - \hat{A})^{-1}\hat{B} + x^{[0]}\hat{C}(\alpha I - \hat{A})^{-2}\hat{B} \end{aligned} \tag{2.15}$$

$$\vdots$$

$$\begin{aligned} &x^{[n-1]}C(\alpha I - A)^{-1}B + \cdots + x^{[0]}C(\alpha I - A)^{-n}B \\ &= x^{[n-1]}\hat{C}(\alpha I - \hat{A})^{-1}\hat{B} + x^{[0]}\hat{C}(\alpha I - \hat{A})^{-n}\hat{B}. \end{aligned} \tag{2.16}$$

Defining the matrix $X \in \mathbb{C}^{n \times np}$ and the generalized observability matrix $\mathcal{O}_{C,A} \in \mathbb{C}^{np \times N}$ as follows:

$$X \doteq \begin{bmatrix} x^{[0]} & & \\ \vdots & \ddots & \\ x^{[n-1]} & \ldots & x^{[0]} \end{bmatrix}; \quad \mathcal{O}_{C,A} \doteq \begin{bmatrix} C(\alpha I - A)^{-1} \\ \vdots \\ C(\alpha I - A)^{-n} \end{bmatrix} \tag{2.17}$$

and defining matrix $\mathcal{O}_{\hat{C},\hat{A}} \in \mathbb{C}^{np \times n}$ analogously by replacing the matrices C and A by $\hat{C}$ and $\hat{A}$ in (2.17), we are able to state the following lemma.

LEMMA 2.2. *A $p \times m$ transfer function $\hat{T}(s) = \hat{C}(sI_n - \hat{A})^{-1}\hat{B}$ satisfies the interpolation conditions* (2.8) *if and only if*

$$X\mathcal{O}_{\hat{C},\hat{A}}\hat{B} = X\mathcal{O}_{C,A}B. \tag{2.18}$$

Proof. Equation (2.18) is simply a matrix form of the system (2.14)–(2.16). ∎

We can transpose the preceding reasoning to the right interpolation condition (2.9). Defining

$$Y = \begin{bmatrix} y^{[0]} & \ldots & y^{[n-1]} \\ & \ddots & \vdots \\ & & y^{[0]} \end{bmatrix}; \quad \mathcal{C}_{A,B} = \begin{bmatrix} (\alpha I - A)^{-1}B \ldots (\alpha I - A)^{-n}B \end{bmatrix} \tag{2.19}$$

and following the same reasoning as before, we obtain the following lemma.

LEMMA 2.3. *A $p \times m$ transfer function $\hat{T}(s) = \hat{C}(sI_n - \hat{A})^{-1}\hat{B}$ verifies the interpolation conditions* (2.9) *if and only if*

$$\hat{C}\mathcal{C}_{\hat{A},\hat{B}}Y = C\mathcal{C}_{A,B}Y. \tag{2.20}$$

At this point, all that we have done is to rewrite the left and right interpolation conditions into matrix equations. Next, we define the generalized Loewner matrix as

$$\mathcal{L}_{T(s)} = X\mathcal{O}_{C,A}\mathcal{C}_{A,B}Y. \tag{2.21}$$

The matrix $\mathcal{L}_{\hat{T}(s)}$ is defined as $\mathcal{L}_{T(s)}$ by replacing the matrices C, A, and B by $\hat{C}, \hat{A}$, and $\hat{B}$. By rewriting the two-sided interpolation conditions corresponding to (2.10), we obtain the following lemma.

LEMMA 2.4. *A $p \times m$ transfer function $\hat{T}(s) = \hat{C}(sI_n - \hat{A})^{-1}\hat{B}$ verifies the interpolation conditions* (2.10) *if and only if*

$$\mathcal{L}_{\hat{T}(s)} = \mathcal{L}_{T(s)}. \tag{2.22}$$

The following result can be proven using partial fraction expansion and Lemmas 2.2 to 2.3.

PROPOSITION 2.5. *Every transfer function $\hat{T}(s)$ that verifies* (2.8), (2.9) *and* (2.10) *is such that*

$$X\mathcal{O}_{C,A}A\mathcal{C}_{A,B}Y = X\mathcal{O}_{\hat{C},\hat{A}}\hat{A}\mathcal{C}_{\hat{A},\hat{B}}Y. \tag{2.23}$$

The main result of this section can now be stated as follows.

PROPOSITION 2.6. *If the matrix $\mathcal{L}_{T(s)}$ is invertible, then every transfer function that verifies the interpolation conditions* (2.8)–(2.10) *has a McMillan degree greater than or equal to n. Moreover, the transfer function of degree n that satisfies the equations* (2.8)–(2.10) *is unique if it exists and it can be constructed by the projection matrices V and Z that satisfy*

$$\mathrm{Im}(V) = \mathrm{Im}\left(\mathcal{C}_{C,A}Y\right), \tag{2.24}$$

$$\mathrm{Ker}(Z^T) = \mathrm{Ker}\left(X\mathcal{O}_{A,B}\right), \tag{2.25}$$

$$Z^TV = I_n, \tag{2.26}$$

if α is not a pole of $\hat{A}$.

Sketch of the proof. Suppose that there exists a transfer function of McMillan degree n such that (2.8)–(2.10) are satisfied. It follows that

$$X\mathcal{O}_{\hat{C},\hat{A}}\hat{B} = X\mathcal{O}_{C,A}B, \tag{2.27}$$

$$\hat{C}\mathcal{C}_{\hat{C},\hat{A}}U = C\mathcal{C}_{C,A}Y, \tag{2.28}$$

$$X\mathcal{O}_{\hat{C},\hat{A}}\hat{A}\mathcal{C}_{\hat{A},\hat{B}}Y = X\mathcal{O}_{C,A}A\mathcal{C}_{A,B}Y. \tag{2.29}$$

Because of the invertibility of $\mathcal{L}_{T(s)}$, the matrices $X\mathcal{O}_{\hat{C},\hat{A}} \in \mathbb{C}^{n\times n}$ and $\mathcal{C}_{\hat{A},\hat{B}}Y \in \mathbb{C}^{n\times n}$ are invertible. If we define

$$M = (X\mathcal{O}_{\hat{C},\hat{A}})^{-1}, \tag{2.30}$$

$$N = (\mathcal{C}_{\hat{A},\hat{B}}Y)^{-1}, \tag{2.31}$$

$$Z^T = MX\mathcal{O}_{C,A}, \tag{2.32}$$

$$V = \mathcal{C}_{A,B}YN, \tag{2.33}$$

it is straightforward to show that

$$\hat{A} = Z^TAV, \quad \hat{B} = Z^TB, \quad \hat{C} = CV, \quad Z^TV = I_n. \tag{2.34}$$

3. Auxiliary results. In this section, we define a generalized Loewner matrix that will allow us to construct explicitly the solution of the interpolation problem (1.1) under some mild conditions. This generalized Loewner matrix is inspired by the discussion in [2]. For the SISO case previous results based on [1], [8], and [10] may be found in [9].

In this section, we are given a strictly proper transfer function $T(s)$ and a $T(s)$-admissible interpolation set $I = \{I_l, I_r\}$ as defined in section 1. The objective of this section is to find a way to characterize the set of strictly proper transfer functions $\hat{T}(s)$ such that I is $\hat{T}(s)$-admissible (the interpolation points are not poles of $\hat{T}(s)$) and $\hat{T}(s)$ tangentially interpolates $T(s)$ at I.

We define first several matrices that will be used in the development. Consider the set I_l and associate with the pair $(z_\alpha, x_\alpha(s)) \in I_l$ defined in (1.11)–(1.12) the matrix $X_\alpha \in \mathbb{C}^{\beta_\alpha \times p\beta_\alpha}$

$$X_\alpha \doteq \begin{bmatrix} x_\alpha^{[0]} & & \\ \vdots & \ddots & \\ x_\alpha^{[\beta_\alpha - 1]} & \dots & x_\alpha^{[0]} \end{bmatrix}, \tag{3.1}$$

and define the matrix $X(I_l) \in \mathbb{C}^{s(I_l) \times ps(I_l)}$ by

$$X(I_l) \doteq \operatorname{diag}\{X_\alpha\}_{\alpha=1}^{k_{left}}. \tag{3.2}$$

Analogously, with the pair $(w_\alpha, y_\alpha(s)) \in I_r$, we associate the matrix

$$Y_\alpha \doteq \begin{bmatrix} y_\alpha^{[0]} & \dots & y_\alpha^{[\delta_\alpha - 1]} \\ & \ddots & \vdots \\ & & y_\alpha^{[0]} \end{bmatrix} \tag{3.3}$$

and define

$$Y(I_r) \doteq \operatorname{diag}\{Y_\alpha\}_{\alpha=1}^{k_{right}} \tag{3.4}$$

related to, respectively, the left and right interpolation sets I_l and I_r.

The Jordan matrices will play an important role in this paper, and we therefore introduce the following compact notation.

DEFINITION 3.1. *The matrix $J_{w,\delta,k} \in \mathbb{C}^{k\delta \times k\delta}$ is defined to be*

$$J_{w,\delta,k} \doteq \begin{bmatrix} wI_k & -I_k & & \\ & \ddots & \ddots & \\ & & \ddots & -I_k \\ & & & wI_k \end{bmatrix}. \tag{3.5}$$

When $k = 1$, $J_{w,\delta,1}$ is simply a Jordan matrix of size $\delta \times \delta$ at eigenvalue w and is written $J_{w,\delta}$.

With this definition, we easily obtain the following lemma.

LEMMA 3.2.

$$J_{w,\delta,m} Y_\alpha = Y_\alpha J_{w,\delta}, \quad J_{w,\beta}^T X_\alpha = X_\alpha J_{w,\beta,p}^T. \tag{3.6}$$

Proof. The case $w = 0$ is nothing but the shift invariance property of block Toeplitz matrices. It then also follows for $J_{w,\delta,m} = wI + J_{0,\delta,m}$ since we add the same term on both sides of (3.6). ∎

Two matrices associated to the $p \times m$ transfer function $T(s) = C(sI_N - A)^{-1}B$ with $A \in \mathbb{C}^{N\times N}$ are the controllability matrix $Contr(A,B) \in \mathbb{C}^{pN\times N}$ and the observability matrix $Obs(C,A) \in \mathbb{C}^{N\times mN}$ defined by

$$Contr(A,B) \doteq \left[B \dots A^{N-1}B\right], \quad Obs(C,A) = \begin{bmatrix} C \\ \vdots \\ CA^{N-1} \end{bmatrix}. \tag{3.7}$$

The quantities occurring in $Contr(A,B)$ and $Obs(C,A)$,

$$\mu_{A,B}(\infty,k) \doteq A^{k-1}B \quad \nu_{C,A}(\infty,k) \doteq CA^{k-1}, \tag{3.8}$$

can be seen as "moments" of $(sI-A)^{-1}B$ and $C(sI-A)^{-1}$ about infinity. Similarly, from the dyadic expansion about a point $\lambda \notin \Lambda(A)$

$$(sI-A)^{-1} = \sum_{k=0}^{+\infty} (\lambda I - A)^{-k-1}(\lambda - s)^k, \tag{3.9}$$

we define the moments about a finite expansion point $\lambda \in \mathbb{C}$

$$\mu_{A,B}(\lambda,k) \doteq (\lambda I - A)^{-k}B, \quad \nu_{C,A}(\lambda,k) \doteq C(\lambda I - A)^{-k}. \tag{3.10}$$

DEFINITION 3.3. *Let I be a $T(s)$-admissible interpolation set. For any state-space realization (A,B,C) of $T(s)$, we associate with the right tangential interpolation set I_r the generalized controllability matrix $\mathcal{C}_{A,B}(I_r)$ by the following equations:*

$$\mathcal{C}_{A,B}(z_\alpha,\beta_\alpha) \doteq \left[\mu(z_\alpha,1) \dots \mu(z_\alpha,\beta_\alpha)\right], \tag{3.11}$$

$$\mathcal{C}_{A,B}(I_r) \doteq \left[\mathcal{C}_{A,B}(z_1,\beta_1) \dots \mathcal{C}_{A,B}(z_{k_{left}},\beta_{k_{left}})\right]. \tag{3.12}$$

Similarly, we define a generalized observability matrix $\mathcal{O}_{C,A}$ with the left tangential interpolation set I_l:

$$\mathcal{O}_{C,A}(w_\alpha,\delta_\alpha) \doteq \begin{bmatrix} \nu(w_\alpha,1) \\ \vdots \\ \nu(w_\alpha,\delta_\alpha) \end{bmatrix}, \mathcal{O}_{C,A}(I_l) \doteq \begin{bmatrix} \mathcal{O}_{C,A}(w_1,\delta_1) \\ \vdots \\ \mathcal{O}_{C,A}(w_{k_{right}},\delta_{k_{right}}) \end{bmatrix}. \tag{3.13}$$

We associate with the tangential interpolation set I the generalized Loewner matrix $\mathcal{L}_{T(s)}(I) \in \mathbb{C}^{s(I_l)\times s(I_r)}$ defined by

$$\mathcal{L}_{T(s)}(I) \doteq X(I_l)\mathcal{O}_{C,A}(I_l)\mathcal{C}_{A,B}(I_r)Y(I_r), \tag{3.14}$$

where (A,B,C) is a minimal realization of $T(s)$.

It is straightforward to verify then that $\mathcal{L}_{T(s)}(I)$ does not depend on the particular state space realization of $T(s)$. Next, we derive a series of lemmas that are needed for our main result in Theorem 3.10.

LEMMA 3.4. *If $z_\alpha \neq w_\gamma$ and both interpolation points are finite,*

$$\begin{aligned}
&\mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)\mathcal{C}_{A,B}(w_\gamma,\delta_\gamma)\\
&= \frac{1}{w_\gamma - z_\alpha}\mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)\left(\begin{bmatrix} B & 0\dots 0\end{bmatrix} - \mathcal{C}_{A,B}(w_\gamma,\delta_\gamma)J_{0,\delta_\gamma,m}\right)\\
&\quad + \frac{1}{z_\alpha - w_\gamma}\left(\begin{bmatrix} C\\ 0\\ \vdots\\ 0\end{bmatrix} - J^T_{0,\beta_\alpha,p}\mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)\right)\mathcal{C}_{A,B}(w_\gamma,\delta_\gamma). \qquad (3.15)
\end{aligned}$$

If $z_\alpha \neq w_\gamma$ and z_α is infinite, then

$$\begin{aligned}
&\mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)\mathcal{C}_{A,B}(w_\gamma,\delta_\gamma)\\
&= \begin{bmatrix} C\\ 0\\ \vdots\\ 0\end{bmatrix}\mathcal{C}_{A,B}(z_\alpha,\delta_\alpha) - J^T_{0,\beta_\alpha}\mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)\mathcal{C}_{A,B}(w_\gamma,\delta_\gamma)J_{0,\delta_\gamma,m} \qquad (3.16)\\
&\quad -w_\gamma J^T_{0,\beta}\mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)\mathcal{C}_{A,B}(w_\gamma,\delta_\gamma) + J^T_{0,\beta_\alpha}\mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)\begin{bmatrix} B & 0\dots 0\end{bmatrix}.
\end{aligned}$$

Proof. We first prove (3.15). Recall that if $\alpha \neq \beta \in \mathbb{C}$, then

$$(\alpha I - A)^{-1}(\beta I - A)^{-1} = \frac{1}{\beta-\alpha}(\alpha I - A)^{-1} + \frac{1}{\alpha-\beta}(\beta I - A)^{-1}. \qquad (3.17)$$

This permits us to write that

$$\begin{aligned}
&\mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)\mathcal{C}_{A,B}(w_\gamma,\delta_\gamma)\\
&= \begin{bmatrix} C(z_\alpha I - A)^{-1}\\ \vdots\\ C(z_\alpha I - A)^{-\beta_\alpha}\end{bmatrix}\left[(w_\gamma I - A)^{-1}B\dots(w_\gamma I - A)^{-\delta_\gamma}B\right] \qquad (3.18)\\
&= \frac{1}{w_\gamma - z_\alpha}\begin{bmatrix} C(z_\alpha I - A)^{-1}\\ \vdots\\ C(z_\alpha I - A)^{-\beta_\alpha}\end{bmatrix}\left[(B\dots(w_\gamma I - A)^{-\delta_\gamma+1}B\right] \qquad (3.19)\\
&\quad + \frac{1}{z_\alpha - w_\gamma}\begin{bmatrix} C\\ \vdots\\ C(z_\alpha I - A)^{-\beta_\alpha+1}\end{bmatrix}\left[(w_\gamma I - A)^{-1}B\dots(w_\gamma I - A)^{-\delta_\gamma}B\right].
\end{aligned}$$

This last equation is equal to (3.15). This concludes the proof for the finite case.

Next, consider the case $z_\alpha = \infty$. The proof is similar but uses the following equality:

$$A(\lambda I - A)^{-1} = -I + \lambda(\lambda I - A)^{-1}. \qquad (3.20)$$

This permits us to write that

$$\begin{aligned}
&\mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)\mathcal{C}_{A,B}(w_\gamma,\delta_\gamma)\\
&= \begin{bmatrix} C \\ \vdots \\ CA^{\beta_\alpha-1} \end{bmatrix} \left[(w_\gamma I - A)^{-1}B \dots (w_\gamma I - A)^{-\delta_\gamma}B\right] \qquad (3.21)\\
&= \begin{bmatrix} C \\ 0 \\ \vdots \\ 0 \end{bmatrix} \mathcal{C}_{A,B}(w_\gamma,\delta_\gamma) - J_{0,\beta_\alpha}^T \mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)(A - w_\gamma I + w_\gamma I)\mathcal{C}_{A,B}(w_\gamma,\delta_\gamma) \qquad (3.22)\\
&= \begin{bmatrix} C \\ 0 \\ \vdots \\ 0 \end{bmatrix} \mathcal{C}_{A,B}(w_\gamma,\delta_\gamma) - w_\gamma J_{0,\beta_\alpha}^T \mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)\mathcal{C}_{A,B}(w_\gamma,\delta_\gamma) \qquad (3.23)\\
&\quad + J_{0,\beta_\alpha}^T \mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)\left(\begin{bmatrix} B & 0 \dots 0 \end{bmatrix} - \mathcal{C}_{A,B}(w_\gamma,\delta_\gamma)J_{0,\delta}\right).
\end{aligned}$$

This last term is equal to the right-hand side of (3.16). ∎

To prove Theorem 3.10, we need the important result that the matrix $\mathcal{L}_{\hat{T}(s)}(I)$ is invariant for any matrix $\hat{T}(s)$ interpolating $T(s)$ at I (for which I is $\hat{T}(s)$-admissible). However, to show this result, we need the following lemmas.

LEMMA 3.5. *Let $T(s) = C(sI - A)^{-1}B$ and $\hat{T}(s) = \hat{C}(sI - \hat{A})^{-1}\hat{B}$ be two $p \times m$ strictly proper transfer functions. Let I_l be a left interpolation set that is $T(s)$- and $\hat{T}(s)$-admissible. Then, $\hat{T}(s)$ interpolates $T(s)$ at I_l if and only if*

$$X(I_l)\mathcal{O}_{\hat{C},\hat{A}}(I_l)\hat{B} = X(I_l)\mathcal{O}_{C,A}(I_l)B. \qquad (3.24)$$

Proof. Because of the diagonal structure of X, if we prove (3.24) for one diagonal block of X, say for instance X_α, we prove it for the entire equation (3.24). So we consider the block associated with X_α, and we drop I_l from $x_\alpha(s), X_\alpha, \mathcal{O}_{C,A}(I_l), \mathcal{O}_{\hat{C},\hat{A}}(I_l)$ to make the notation simpler. In other words, we consider the case where there is only one vector $x(s)$ of degree $\beta - 1$ associated with one interpolation point z in the left interpolation set I_l. We assume that z is finite (appropriate change must be made for the case $z = \infty$). We have to show that (1.5) is satisfied if and only if

$$X\mathcal{O}_{\hat{C},\hat{A}}\hat{B} = X\mathcal{O}_{C,A}B. \qquad (3.25)$$

We can write that

$$T(s) = \sum_{i=0}^{+\infty} C(zI - A)^{-i-1}B(z - s)^i, \quad \hat{T}(s) = \sum_{i=0}^{+\infty} \hat{C}(zI - \hat{A})^{-i-1}\hat{B}(z - s)^i. \qquad (3.26)$$

Equation (1.5) says that $x(s)$ is a left zero of $T(s) - \hat{T}(s)$. This means that the first β Taylor coefficients of $x(s)(T(s) - \hat{T}(s))$ at $s = z$ are zero. In other words, for all $1 \le i \le \beta$, the following equation must be satisfied:

$$\sum_{k=0}^{i-1} x^{[k]}\hat{C}(zI - \hat{A})^{i-k}\hat{B} = \sum_{k=0}^{i-1} x^{[k]}C(zI - A)^{i-k}B, \qquad (3.27)$$

and this equation turns out to be exactly the ith row of (3.25). ∎

Analogously, for the right interpolation conditions, we have the following lemma.

LEMMA 3.6. *Let $T(s) = C(sI - A)^{-1}B$ and $\hat{T}(s) = \hat{C}(sI - \hat{A})^{-1}\hat{B}$ be two $p \times m$ strictly proper transfer functions. Let I_r be a right interpolation set that is $T(s)$- and $\hat{T}(s)$-admissible. Then, $\hat{T}(s)$ interpolates $T(s)$ at I_r if and only if*

$$\hat{C}\mathcal{C}_{\hat{A},\hat{B}}Y = C\mathcal{C}_{A,B}Y. \tag{3.28}$$

The proof is similar to the proof of Lemma 3.5.

LEMMA 3.7. *Let $T(s) = C(sI - A)^{-1}B$ and $\hat{T}(s) = \hat{C}(sI - \hat{A})^{-1}\hat{B}$ be two $p \times m$ strictly proper transfer functions. Let $I = \{I_l, I_r\}$ be an interpolation set that is $T(s)$- and $\hat{T}(s)$-admissible. If $\hat{T}(s)$ interpolates $T(s)$ at I and then, for every pair of indices α, γ such that $z_\alpha = w_\gamma = \xi$, (where ξ is finite),*

$$X_\alpha \mathcal{O}_{\hat{C},\hat{A}}(z_\alpha, \beta_\alpha)\mathcal{C}_{\hat{A},\hat{B}}(w_\gamma, \delta_\gamma)Y_\gamma = X_\alpha \mathcal{O}_{C,A}(z_\alpha, \beta_\alpha)\mathcal{C}_{A,\hat{B}}(w_\gamma, \delta_\gamma)Y_\gamma; \tag{3.29}$$

and for every pair of indices α, γ such that $z_\alpha = w_\gamma = \xi$, (where $\xi = \infty$),

$$X_\alpha \mathcal{O}_{\hat{C},\hat{A}}(z_\alpha, \beta_\alpha)\hat{A}\mathcal{C}_{\hat{A},\hat{B}}(w_\gamma, \delta_\gamma)Y_\gamma = X_\alpha \mathcal{O}_{C,A}(z_\alpha, \beta_\alpha)A\mathcal{C}_{A,B}(w_\gamma, \delta_\gamma)Y_\gamma. \tag{3.30}$$

Proof. We consider the finite case. To simplify the notation, we drop the subscripts α, γ. Let us choose two integers f, g such that $1 \le f \le \beta$ and $1 \le g \le \delta$. Condition 3 of Definition 1.3 applied to $x(s) = x_\alpha^{(f)}(s)$ and $y(s) = y_\gamma^{(g)}(s)$ says that the $f + g$ first derivatives of $x^{(f)}(s)(T(s) - \hat{T}(s))y^{(g)}(s)$ at $s = \xi$ are zero. The condition corresponding to the derivative of highest order is

$$\frac{1}{(f+g-1)!}\frac{d^{f+g-1}}{ds^{f+g-1}}\left\{x^{(f)}(s)\hat{T}(s)y^{(g)}(s)\right\}\Big|_{s=\xi}$$

$$= \sum_{k=0}^{f-1}\sum_{l=0}^{g-1} x^{[k]}C(\xi I - A)^{k+l-f-g}By^{[l]} \tag{3.31}$$

$$= \sum_{k=0}^{f-1}\sum_{l=0}^{g-1}(x^{[k]}C(\xi I - A)^{k-f})((\xi I - A)^{l-g}Bu^{[l]}) \tag{3.32}$$

$$= \left(X\mathcal{O}_{C,A}\mathcal{C}_{A,B}Y\right)_{f,g}. \tag{3.33}$$

Thus, (3.29) is a consequence of the interpolation conditions. The proof is similar for the infinite interpolation point. ∎

Equations (3.25), (2.20), (3.29), and (3.30) are just a matrix version of the interpolation conditions of Definition 1.3. We now proceed to prove that (3.25) and (2.20) imply as well that $X\mathcal{O}_{\hat{C},\hat{A}}\mathcal{C}_{\hat{A},\hat{B}}Y = X\mathcal{O}_{C,A}\mathcal{C}_{A,B}Y$ and $X\mathcal{O}_{\hat{C},\hat{A}}\hat{A}\mathcal{C}_{\hat{A},\hat{B}}Y = X\mathcal{O}_{C,A}A\mathcal{C}_{A,B}Y$, provided the two-sided interpolation condition 3 of Definition 1.3 is added for every pair $z_\alpha = w_\gamma$. This may seem surprising but it is a simple consequence of Lemma 3.7 when $z_{\alpha \ne w_\gamma}$ and follows from the two-sided condition when $z_\alpha = w_\alpha$.

LEMMA 3.8. *If the strictly proper transfer function $\hat{T}(s) = \hat{C}(sI - \hat{A})^{-1}\hat{B}$ interpolates $T(s)$ at $I = \{I_l, I_r\}$ (where the interpolation set I is $T(s)$- and $\hat{T}(s)$-admissible), then*

$$X\mathcal{O}_{\hat{C},\hat{A}}\mathcal{C}_{\hat{A},\hat{B}}Y = X\mathcal{O}_{C,A}\mathcal{C}_{A,B}Y. \tag{3.34}$$

Proof. The proof will be done block by block. If $z_\alpha = w_\gamma = \xi_{\alpha,\gamma}$ and $\xi_{\alpha,\gamma}$ is finite, the proof follows from Lemma 3.7. Let us consider the case $\xi_{\alpha,\gamma}$ infinite.

$$
\begin{aligned}
& X_\alpha \mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)\mathcal{C}_{A,B}(w_\gamma,\delta_\gamma)Y_\gamma \\
& = X_\alpha \mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)\left[B \dots A^{\delta_\gamma-1}\right]
\begin{bmatrix} y_\gamma^{[0]} & \dots & y_\gamma^{\delta_\gamma-1} \\ & \ddots & \vdots \\ & & y_\gamma^{[0]} \end{bmatrix} && (3.35)\\
& = X_\alpha \mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)B\left[y_\gamma^{[0]} \dots y_\gamma^{\delta_\gamma-1}\right] && (3.36)\\
& \quad -X_\alpha \mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)A\mathcal{C}_{A,B}(w_\gamma,\delta_\gamma)Y_\gamma J_{0,\delta} && (3.37)\\
& = X_\alpha \mathcal{O}_{\hat{C},\hat{A}}(z_\alpha,\beta_\alpha)\mathcal{C}_{\hat{A},\hat{B}}(w_\gamma,\delta_\gamma)Y_\gamma. && (3.38)
\end{aligned}
$$

Second, we suppose that

$$z_\alpha \neq w_\gamma. \tag{3.39}$$

We assume that z_α and w_γ are finite. The idea is to recursively use (3.15). We want to show that

$$X_\alpha \mathcal{O}_{\hat{C},\hat{A}}(z_\alpha,\beta_\alpha)\hat{B} = X_\alpha \mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)B \tag{3.40}$$

and

$$\hat{C}\mathcal{C}_{\hat{A},\hat{B}}(w_\gamma,\delta_\gamma)Y_\gamma = C\mathcal{C}_{A,B}(w_\gamma,\delta_\gamma)Y_\gamma \tag{3.41}$$

imply

$$X_\alpha \mathcal{O}_{\hat{C},\hat{A}}(z_\alpha,\beta_\alpha)\mathcal{C}_{\hat{A},\hat{B}}(w_\gamma,\delta_\gamma)Y_\gamma = X_\alpha \mathcal{O}_{C,A}(z_\alpha,\beta_\alpha)\mathcal{C}_{A,B}(w_\gamma,\delta_\gamma)Y_\gamma. \tag{3.42}$$

We drop again $\alpha,\gamma,(z_\alpha,\beta_\alpha),(w_\gamma,\delta_\gamma)$ to simplify the notation.

$$
\begin{aligned}
X\mathcal{O}_{C,A}\mathcal{C}_{A,B}Y &= \frac{1}{w-z}X\mathcal{O}_{C,A}\left(\begin{bmatrix} B & 0\dots 0\end{bmatrix} - \mathcal{C}_{A,B}J_{0,\delta,m}\right)Y \\
&\quad + \frac{1}{z-w}X\left(\begin{bmatrix} C \\ 0 \\ \vdots \\ 0 \end{bmatrix} - J_{0,\beta,p}^T\mathcal{O}_{C,A}\right)\mathcal{C}_{A,B}Y && (3.43)\\
&= \frac{1}{w-z}\begin{bmatrix} X\mathcal{O}_{C,A}B & 0\dots 0\end{bmatrix}Y + \frac{1}{z-w}X\begin{bmatrix} C\mathcal{C}_{A,B}Y \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\
&\quad - \frac{1}{w-z}X\mathcal{O}_{C,A}\mathcal{C}_{A,B}YJ_{0,\delta} - \frac{1}{z-w}J_{0,\beta}X\mathcal{O}_{C,A}\mathcal{C}_{A,B}Y. && (3.44)
\end{aligned}
$$

From Lemmas 3.5 and 3.6 we deduce

$$\frac{1}{w-z}\begin{bmatrix} X\mathcal{O}_{\hat{C},\hat{A}}\hat{B} & 0\dots 0\end{bmatrix}Y = \frac{1}{w-z}\begin{bmatrix} X\mathcal{O}_{\hat{C},\hat{A}}\hat{B} & 0\dots 0\end{bmatrix}Y, \tag{3.45}$$

$$\frac{1}{z-w}X\begin{bmatrix} C\mathcal{C}_{A,B}Y \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \frac{1}{z-w}X\begin{bmatrix} \hat{C}\mathcal{C}_{\hat{A},\hat{B}}Y \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{3.46}$$

By using a recursive argument, it can be shown that

$$X\mathcal{O}_{C,A}\mathcal{C}_{A,B}YJ_{0,\delta} = X\mathcal{O}_{\hat{C},\hat{A}}\mathcal{C}_{\hat{A},\hat{B}}YJ_{0,\delta}, \tag{3.47}$$

$$J_{0,\beta}X\mathcal{O}_{C,A}\mathcal{C}_{A,B}Y = J_{0,\beta}X\mathcal{O}_{\hat{C},\hat{A}}\mathcal{C}_{\hat{A},\hat{B}}Y. \tag{3.48}$$

Finally, we have to consider the case with one infinite interpolation point, say for instance $z_\alpha = \infty$ and the other point w_γ finite. This can be treated similarly by recursively using (3.16). □

LEMMA 3.9. *If the strictly proper transfer function $\hat{T}(s) = \hat{C}(sI - \hat{A})^{-1}\hat{B}$ interpolates $T(s)$ at $I = \{I_l, I_r\}$ and I is $T(s)$- and $\hat{T}(s)$-admissible, then*

$$X\mathcal{O}_{\hat{C},\hat{A}}\hat{A}\mathcal{C}_{\hat{A},\hat{B}}Y = X\mathcal{O}_{C,A}A\mathcal{C}_{A,B}Y. \tag{3.49}$$

Proof. We recall that

$$A\mathcal{C}_{A,B}Y = \left[A\mathcal{C}_{A,B}(w_1,\delta_1)Y_1 \dots A\mathcal{C}_{A,B}(w_s,\delta_s)Y_s\right]. \tag{3.50}$$

The proof will again be done block by block. Let us prove it for the block of $\mathcal{C}_{\hat{A},\hat{B}}(I_r)Y$ corresponding to w_γ. Two cases must be considered.

Assuming that w_γ is finite yields

$$\begin{aligned}
&A\mathcal{C}_{C,A}(w_\gamma,\delta_\gamma)Y_\gamma \\
&= (A - w_\gamma I + w_\gamma I)\mathcal{C}_{C,A}(w_\gamma,\delta_\gamma)Y_\gamma && (3.51)\\
&= -\left[B \dots (w_\gamma I_N - A)^{-\delta_\gamma+1}B\right]Y_\gamma + w_\gamma\mathcal{C}_{C,A}(w_\gamma,\delta_\gamma)Y_\gamma && (3.52)\\
&= -B\left[y^{[0]} \dots y^{[\delta_\gamma-1]}\right] + \mathcal{C}_{C,A}(w_\gamma,\delta_\gamma)Y_\gamma J_{w_\gamma,\delta_\gamma}. && (3.53)
\end{aligned}$$

This allows us to write that

$$\begin{aligned}
&X\mathcal{O}_{\hat{C},\hat{A}}\hat{A}\mathcal{C}_{\hat{A},\hat{B}}(w_\gamma,\delta_\gamma)Y_\gamma \\
&= X\mathcal{O}_{\hat{C},\hat{A}}(-\hat{B}[y^{[0]} \ \dots \ y^{[\delta_\gamma-1]}] + \mathcal{C}_{\hat{A},\hat{B}}(w_\gamma,\delta_\gamma)Y_\gamma J_{w_\gamma,\delta_\gamma}) && (3.54)\\
&= X\mathcal{O}_{C,A}(-B[y^{[\delta_\gamma-1]} \ \dots \ y^{[0]}] + \mathcal{C}_{A,B}(w_\gamma,\delta_\gamma)Y_\gamma J_{w_\gamma,\delta_\gamma}) && (3.55)\\
&= X\mathcal{O}_{C,A}A\mathcal{C}_{A,B}(w_\gamma,\delta_\gamma)Y_\gamma, && (3.56)
\end{aligned}$$

where the first part of (3.55) is a consequence of Lemma 3.5 and the second part of (3.55) is a consequence of Lemma 3.7.

Second, assume that $w_\gamma = \infty$. Two cases must be considered. If z_α is finite, then the proof is done by transposing the preceding results. If $\xi_{\alpha,\gamma} = \infty$, then this follows from Lemma 3.7. □

Putting together the preceding results, we obtain the following theorem that gives the main result of the section.

THEOREM 3.10. *Let (C_1, A_1, B_1) be a minimal state space realization of the strictly proper transfer function $T_1(s)$ and (C_2, A_2, B_2) be a minimal state space realization of the strictly proper transfer function $T_2(s)$. Let the interpolation set $I = \{I_l, I_r\}$ be $T_1(s)$- and $T_2(s)$-admissible (i.e., the interpolation points are neither poles of $T_1(s)$ nor $T_2(s)$). Then, $T_1(s)$ interpolates $T_2(s)$ at I if and only if the following equations are satisfied:*

$$C_1\mathcal{C}_{A_1,B_1}(I_r)Y(I_r) = C_2\mathcal{C}_{A_2,B_2}(I_r)Y(I_r), \tag{3.57}$$

$$X(I_l)\mathcal{O}_{C_1,A_1}(I_l)B_1 = X(I_l)\mathcal{O}_{C_2,A_2}(I_l)B_2, \tag{3.58}$$

$$X(I_l)\mathcal{O}_{C_1,A_1}(I_l)\mathcal{C}_{A_1,B_1}(I_r)Y(I_r) = X(I_l)\mathcal{O}_{C_2,A_2}(I_l)\mathcal{C}_{A_2,B_2}(I_r)Y(I_r), \tag{3.59}$$

$$X(I_l)\mathcal{O}_{C_1,A_1}(I_l)A_1\mathcal{C}_{A_1,B_1}(I_r)Y(I_r) = X(I_l)\mathcal{O}_{C_2,A_2}(I_l)A_2\mathcal{C}_{A_2,B_2}(I_r)Y(I_r). \tag{3.60}$$

Proof. The proof follows from the preceding results. ∎

4. The multipoint Padé reduced order transfer function. In this section, we give a practical way of constructing a minimal state space realization of the transfer function of minimal McMillan degree that interpolates $T(s)$ at the interpolation set I when the corresponding Loewner matrix $\mathcal{L}_{T(s)}(I)$ is invertible. The interpolating transfer function of minimal McMillan degree will be called the multipoint Padé reduced order transfer function $\hat{T}_{MP}(s)$. A minimal state space realization $(\hat{C}_{MP}, \hat{A}_{MP}, \hat{B}_{MP})$ of $\hat{T}_{MP}(s)$ will be obtained by a projection technique. More precisely, the state space realization $(\hat{C}_{MP}, \hat{A}_{MP}, \hat{B}_{MP})$ will be constructed by projecting a minimal state space realization (C, A, B) of $T(s)$ with two projecting matrices $Z, V \in \mathbb{C}^{N\times n}$ as follows:

$$\hat{C}_{MP} = CV, \quad \hat{A}_{MP} = Z^T AV, \quad \hat{B}_{MP} = Z^T B, \quad Z^T V = I_n.$$

It will be shown that the projecting matrices Z, V can be obtained by solving Sylvester equations.

In order to prove these facts, we first introduce two new pairs of matrices. Let us consider the left tangential interpolation set I_l defined in (1.13). For any integer α such that $1 \le \alpha \le k_{left}$, define the matrices $(L_\alpha^{(l)}, L_\alpha^{(r)})$ as follows:

1. If the interpolation point z_α is finite, then take

$$L_\alpha^{(l)} \doteq I_{\beta_\alpha}, \quad L_\alpha^{(r)} \doteq J_{z_\alpha,\beta_\alpha}^T. \tag{4.1}$$

2. If the interpolation point z_α is infinite, then define

$$L_\alpha^{(l)} \doteq -J_{0,\beta_\alpha}^T, \quad L_\alpha^{(r)} \doteq I_{\beta_\alpha}. \tag{4.2}$$

Moreover, define the matrix $\mathcal{X}_\alpha$ as follows:

$$\mathcal{X}_\alpha = \begin{bmatrix} x_\alpha^{[0]} \\ \vdots \\ x_\alpha^{[\beta_\alpha - 1]} \end{bmatrix}. \tag{4.3}$$

Finally, define the matrices $L^{(l)}(I_l)$, $L^{(r)}(I_l)$, and $\mathcal{X}(I_l)$ as follows:

$$L^{(l)}(I_l) \doteq \operatorname{diag}\{L_\alpha^{(l)}\}_{\alpha=1}^{k_{left}}, \quad L^{(r)}(I_l) \doteq \operatorname{diag}\{L_\alpha^{(r)}\}_{\alpha=1}^{k_{left}}, \tag{4.4}$$

$$\mathcal{X}(I_l) \doteq \begin{bmatrix} \mathcal{X}_1 \\ \vdots \\ \mathcal{X}_{k_{left}} \end{bmatrix}. \tag{4.5}$$

Let us consider the right tangential interpolation set I_r defined in (1.16). For any integer α such that $1 \le \alpha \le k_{right}$, define the matrices $(R_\alpha^{(l)}, R_\alpha^{(r)})$ as follows:

1. If the interpolation point w_α is finite, then take

$$R_\alpha^{(l)} \doteq I_{\delta_\alpha}, \quad R_\alpha^{(r)} \doteq J_{w_\alpha,\delta_\alpha}. \tag{4.6}$$

2. If the interpolation point w_α is infinite, then define

$$R_\alpha^{(l)} \doteq -J_{0,\delta_\alpha}, \quad R_\alpha^{(r)} \doteq I_{\delta_\alpha}. \tag{4.7}$$

Moreover, define

$$\mathcal{Y}_\alpha \doteq \left[y_\alpha^{[0]} \dots y_\alpha^{[\delta_\alpha - 1]}\right]. \tag{4.8}$$

Finally, define the matrices $R^{(l)}(I_r)$, $R^{(r)}(I_r)$, and $\mathcal{Y}(I_r)$ as follows:

$$R^{(l)}(I_r) \doteq \mathrm{diag}\{R_\alpha^{(l)}\}_{\alpha=1}^{k_{right}} \ , \ R^{(r)}(I_r) \doteq \mathrm{diag}\{R_\alpha^{(r)}\}_{\alpha=1}^{k_{right}} \tag{4.9}$$

$$\mathcal{Y}(I_r) \doteq \left[\mathcal{Y}_0 \dots \mathcal{Y}_{k_{right}}\right]. \tag{4.10}$$

As a consequence of these definitions we have

$$L^{(l)}L^{(r)} = L^{(r)}L^{(l)}, \quad R^{(l)}R^{(r)} = R^{(r)}R^{(l)} \tag{4.11}$$

and we can now derive the following lemma that introduces the related Sylvester equations.

LEMMA 4.1. *Let (A, B, C) be a state-space realization of the transfer function $T(s)$. Let us consider a $T(s)$-admissible interpolation set $I = \{I_l, I_r\}$. Then,*

$$N = \mathcal{C}_{A,B}(I_r)Y(I_r) \Longleftrightarrow ANR^{(l)}(I_r) - NR^{(r)}(I_r) + B\mathcal{Y}(I_r) = 0, \tag{4.12}$$

$$M = X(I_l)\mathcal{O}_{C,A}(I_l) \Longleftrightarrow L^{(l)}(I_l)MA - L^{(r)}M + \mathcal{X}C = 0. \tag{4.13}$$

Proof. Let us prove (4.12) for only one interpolation condition $I_r = \{(w, y(s))\}$ at a finite point w.

$$\begin{aligned} & ANR^{(l)}(I_r) - NR^{(r)}(I_r) + B\mathcal{Y}(I_r) = 0 \\ \Longleftrightarrow\ & A\left[n_1 \dots n_k\right] - \left[n_1 \dots n_k\right] J_{w,k} \\ & +B\left[y^{[0]} \dots y^{[k-1]}\right] = 0. \end{aligned} \tag{4.14}$$

Let us solve this linear equation for N column by column from n_1 up to n_k. We find recursively that

$$(wI - A)n_1 = By^{[0]} \tag{4.15}$$

$$(wI - A)n_{i+1} = By^{[i]} + n_i. \tag{4.16}$$

Moreover, the matrix $wI - A$ is invertible because we always assume here that the interpolation set I is $T(s)$-admissible. This proves that $N = \mathcal{C}_{A,B}(I_r)Y(I_r)$ for one finite interpolation condition $I_r = \{(w, y(s))\}$.

Let us prove (4.12) for only one interpolation condition $I_r = \{(w, y(s))\}$ at an infinite point $w = \infty$.

$$\begin{aligned} & ANR^{(l)}(I_r) - NR^{(r)}(I_r) + B\mathcal{Y}(I_r) = 0 \\ \Longleftrightarrow\ & A\left[n_1 \dots n_k\right] \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & & & \ddots & 1 \\ 0 & \dots & \dots & \dots & 0 \end{bmatrix} \\ & - \left[n_1 \dots n_k\right] + B\left[y^{[0]} \dots y^{[k-1]}\right] = 0. \end{aligned} \tag{4.17}$$

Again, by solving this equation column by column we find that $N = \mathcal{C}_{A,B}(I_r)Y(I_r)$ for one interpolation condition $I_r = \{(\infty, y(s))\}$. If the interpolation set I_r contains more than one pair, say k_r pairs, because of the block diagonal structure of $R^{(l)}, R^{(r)}$ and $Y(I_r)$, and the block structure of $\mathcal{Y}(I_r)$, we can split the columns of N into k_r blocks and prove the result for each pair $(w_\gamma, y_\gamma(s)) \in I_r$ in order to prove that

$$\begin{aligned} N &= \begin{bmatrix} N_1 \dots N_{k_r} \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{C}_{A,B}\left(w_1, y_1(s)\right) Y\left(w_1, y_1(s)\right) \dots \mathcal{C}_{A,B}\left(w_{k_r}, y_{k_r}(s)\right) Y\left(w_{k_r}, y_{k_r}(s)\right) \end{bmatrix} \\ &= \mathcal{C}_{A,B}(I_r)Y(I_r). \qquad \square \end{aligned} \tag{4.18}$$

The main result of this paper can now be formalized.

THEOREM 4.2. *Consider a transfer function $T(s)$ and a $T(s)$-admissible tangential interpolation set I and assume that the corresponding Loewner matrix $\mathcal{L}_{T(s)}(I) \in \mathbb{C}^{n\times n}$ is invertible. Define then two invertible matrices $M, N \in \mathbb{C}^{n\times n}$ such that*

$$\mathcal{L}_{T(s)} \doteq X\mathcal{O}_{C,A}\mathcal{C}_{A,B}Y = MN, \tag{4.19}$$

and define the "multipoint Padé" reduced order transfer function $\hat{T}_{MP}(s)$ via its state space realization $\{\hat{A}_{MP}, \hat{B}_{MP}, \hat{C}_{MP}\}$ given by the equations

$$\hat{C}_{MP}N = C\mathcal{C}_{A,B}Y, \tag{4.20}$$

$$M\hat{B}_{MP} = X\mathcal{O}_{C,A}B, \tag{4.21}$$

$$M\hat{A}_{MP}N = X\mathcal{O}_{C,A}A\mathcal{C}_{A,B}Y. \tag{4.22}$$

If the interpolation points are not poles of $\hat{T}_{MP}(s)$, i.e., if the interpolation set I is $\hat{T}_{MP}(s)$-admissible, then $\hat{T}_{MP}(s)$ interpolates $T(s)$ at I. Moreover, $\hat{T}_{MP}(s)$ is the unique transfer function of McMillan degree $s(I_l) = s(I_r)$ that interpolates $T(s)$ at I and there exists no such transfer function of lower McMillan degree.

Proof. First, note that it is always possible to find a couple of invertible matrices M, N that satisfy (4.19) because of the invertibility of $\mathcal{L}_{T(s)}(I)$. Second, it can be verified that $\hat{T}_{MP}(s)$ is uniquely defined and does not depend on the particular choice of matrices M, N satisfying (4.19).

The proof consists of showing that $M = X(I_l)\mathcal{O}_{\hat{C}_{MP},\hat{A}_{MP}}(I_l)$ and that $N = \mathcal{C}_{\hat{A}_{MP},\hat{B}_{MP}}(I_r)Y(I_r)$. From the preceding results, it is equivalent to show that M and N are solutions of the Sylvester equations of Lemma 4.1. First, from (4.19)–(4.22) and Lemma 4.1, we have

$$\begin{aligned} &\hat{A}_{MP}NR^{(l)} - NR^{(r)} + \hat{B}_{MP}\mathcal{Y} \\ &= M^{-1}X\mathcal{O}_{C,A}(A\mathcal{C}_{A,B}YR^{(l)} - \mathcal{C}_{A,B}YR^{(r)} + B\mathcal{Y}) = 0. \end{aligned} \tag{4.23}$$

This implies also from Lemma 4.1 that $N = \mathcal{C}_{\hat{A}_{MP},\hat{B}_{MP}}(I_r)Y(I_r)$. Analogously, $M = X(I_l)\mathcal{O}_{\hat{C}_{MP},\hat{A}_{MP}}(I_l)$. The proof follows now from Proposition 3.10. Indeed, (4.20) is equivalent to saying that the right tangential interpolation conditions are satisfied, (4.21) corresponds to the left tangential equations and (4.19) and (4.22) are equivalent to the two-sided interpolation conditions. Hence, $\hat{T}_{MP}(s)$ interpolates $T(s)$ at I.

We have still to prove that $\hat{T}_{MP}(s)$ is the unique transfer function of McMillan degree n that satisfies the interpolation conditions with respect to $T(s)$, and that there exist no transfer function of McMillan degree smaller than n that satisfies the interpolation conditions. To do this, first assume that there exists $\hat{T}(s)$ of McMillan

degree $k < n$ that satisfies the interpolation conditions. Let $(\hat{C}, \hat{A}, \hat{B})$ be a minimal state space realization of $\hat{T}(s)$. Clearly,

$$\operatorname{rank} \mathcal{C}_{\hat{A},\hat{B}}(I_r) Y(I_r) \leq \operatorname{rank} \mathcal{C}_{\hat{A},\hat{B}}(I_r) = \operatorname{rank} Contr(\hat{A}, \hat{B}) = k < n. \tag{4.24}$$

From the interpolation conditions, we must have that $\mathcal{L}_{T(s)}(I) = \mathcal{L}_{\hat{T}(s)}(I)$. This implies that

$$n = \operatorname{rank} \mathcal{L}_{T(s)}(I) = \operatorname{rank} \mathcal{L}_{\hat{T}(s)}(I) \leq k. \tag{4.25}$$

This proves that it is not possible to find an interpolating transfer function of McMillan degree smaller than n.

If we assume that there exists another interpolating transfer function $\hat{T}(s)$ of McMillan degree n, it is not difficult to verify that the procedure given for constructing a minimal state space realization $(\hat{C}, \hat{A}, \hat{B})$ of $\hat{T}(s)$ will produce a state space realization that is similar to $(\hat{C}_{MP}, \hat{A}_{MP}, \hat{B}_{MP})$. This implies that $\hat{T}(s) = \hat{T}_{MP}(s)$ and concludes the proof. □

By inverting the matrices M and N into (4.19)–(4.22), if we define

$$Z^T = M^{-1} X \mathcal{O}_{C,A}, \quad V = \mathcal{C}_{A,B} Y N^{-1}, \tag{4.26}$$

we see that

$$Z^T V = I_n, \quad CV = \hat{C}_{MP}, \quad Z^T B = \hat{B}_{MP}, \quad Z^T A V = \hat{A}_{MP}. \tag{4.27}$$

5. Concluding remarks. An important result that has not been considered in this paper is the following. Assume that a reduced order transfer function $\hat{T}_1(s)$ has been constructed that interpolates the original transfer function $T(s)$ at the interpolation set I_1 with the projecting matrices Z_1 and V_1. If one wants to add new interpolation conditions, say I_2, all that we have to do is to compute the generalized Krylov subspaces corresponding to the new interpolation set I_2 and to construct new projecting matrices Z_2, V_2 that contain, respectively, the column span of Z_1 and V_1 and the new, respectively, left and right generalized Krylov subspaces.

Another important result that can easily be derived is that we only need the projecting matrices Z, V to contain some subspaces, but they can contain other subspaces as well! For instance, Theorem 4.2 can be generalized as follows.

THEOREM 5.1. *Consider a transfer function $T(s) \doteq C(sI - A)^{-1}B$ and a $T(s)$-admissible tangential interpolation set $I \doteq \{I_l, I_r\}$. Let us assume that the projecting matrices Z, V (such that $Z^T V = I_n$) are such that*

$$\begin{aligned} Colsp(V) &\supseteq Colsp\left(\mathcal{C}_{A,B}(I_r) Y(I_r)\right), \\ Colsp(Z^T) &\supseteq Colsp\left(\mathcal{O}^T_{C,A}(I_l) X^T(I_l)\right). \end{aligned}$$

Then, if the interpolation point of I are not poles of $\hat{T}(s) \doteq CV(sI_n - Z^T A V)^{-1} Z^T B$, the transfer function $\hat{T}(s)$ interpolates $T(s)$ at I.

It should also be pointed out that this Krylov technique can easily be extended to generalized state space systems, also called descriptor systems.

Finally, we have shown that the projecting matrices Z, V, constructed in order to compute a state space realization of $\hat{T}_{MP}(s)$, are solutions of Sylvester equations. Actually, it can be shown that, generically, constructing a reduced order transfer function with projecting matrices that are solutions of a Sylvester equation with

respect to a state space realization of the original transfer function is equivalent to solving a particular tangential interpolation problem. We refer to [10] for results in this direction.

Appendix.

LEMMA A.1. *Let $T(s)$ and $\hat{T}(s)$ be two* strictly proper *$p \times m$ transfer functions. $\hat{T}(s)$ tangentially interpolates $T(s)$ at I with respect to Definition* 1.3 *if and only if the three following conditions are satisfied:*
for all finite $z_\alpha, 1 \le \alpha \le r$, for any $1 \le i \le \beta_\alpha$:

$$\frac{d^{i-1}}{ds^{i-1}} \left\{ x_\alpha(s)(T(s) - \hat{T}(s)) \right\} \Big|_{s=z_\alpha} = 0; \tag{A.1}$$

for all $z_\alpha = \infty, 1 \le \alpha \le r$,

$$x_\alpha(s)(T(s) - \hat{T}(s)) = O(s^{-1})^{\beta_\alpha + 1}; \tag{A.2}$$

for all finite w_α, $1 \le \alpha \le s$, for any $1 \le i \le \delta_\alpha$,

$$\frac{d^{i-1}}{ds^{i-1}} \left\{ (T(s) - \hat{T}(s)) y_\alpha(s) \right\} \Big|_{s=w_\alpha} = 0; \tag{A.3}$$

for all $w_\alpha = \infty, 1 \le \alpha \le s$,

$$(T(s) - \hat{T}(s)) y_\alpha(s) = O(s^{-1})^{\delta_\alpha + 1}; \tag{A.4}$$

for all finite $\xi_{\alpha,\gamma}$, for all $f = 1, \ldots, \beta_\alpha$, $g = 1, \ldots, \delta_\gamma$,

$$\frac{d^{f+g-1}}{ds^{f+g-1}} \left\{ x_\alpha^{(f)}(s)(T(s) - \hat{T}(s)) y_\gamma^{(g)}(s) \right\} \Big|_{s=\xi_{\alpha,\gamma}} = 0; \tag{A.5}$$

for all infinite $\xi_{\alpha,\gamma}$, the coefficient $e^{[f+g]}$ of s^{-f-g} of the product

$$x_\alpha^{(f)}(s)(T(s) - \hat{T}(s)) y_\gamma^{(g)}(s) \doteq \sum_{k=1}^{+\infty} e^{[k]} s^{-k} \tag{A.6}$$

is zero, where $f = 1, \ldots, \beta_\alpha$; $g = 1, \ldots, \delta_\gamma$.

Proof of Lemma A.1. It is easy to see that the left tangential interpolation conditions (A.1)–(A.2) and condition 1 of Definition 1.3 are equivalent. For the same reasons, the right tangential interpolation conditions (A.3)–(A.4) and conditions 2 of Definition 1.3 are equivalent. Moreover, it is not difficult to see that the two-sided tangential interpolation condition 3 of Definition 1.3 implies conditions (A.5)–(A.6). The proof will be completed by showing that conditions (A.1)–(A.6) imply conditions 1, 2, and 3 of Definition 1.3.

Let us first consider the case with a finite left and right interpolation point $z \in \mathbb{C}$. As usual, we assume that this point is admissible for $T(s)$ and $\hat{T}(s)$; i.e., it is neither a pole of $T(s)$ nor a pole of $\hat{T}(s)$. So, we assume that we are given two polynomial vectors $x(s)$ and $y(s)$ of respective degree $\beta - 1$ and $\delta - 1$ such that

$$x(s)(T(s) - \hat{T}(s)) = O(s - z)^{\beta}, \quad x(z) \neq 0, \tag{A.7}$$

$$(T(s) - \hat{T}(s)) y(s) = O(s - z)^{\delta}, \quad y(z) \neq 0, \tag{A.8}$$

and for all $1 \le f \le \beta$, $1 \le g \le \delta$,

$$\frac{d^{f+g-1}}{ds^{f+g-1}} \left| \left\{ x^{(f)}(s)(T(s) - \hat{T}(s))y^{(g)}(s) \right\} \right|_{s=z} = 0. \tag{A.9}$$

We want to prove that this implies for all $1 \le f \le \beta$, $1 \le g \le \delta$,

$$x^{(f)}(s)(T(s) - \hat{T}(s))y^{(g)}(s) = O(s-z)^{f+g}. \tag{A.10}$$

By using Lemma 3.7, (A.10) is equivalent to the equation

$$X\mathcal{O}_{C,A}\mathcal{C}_{A,B}Y = X\mathcal{O}_{\hat{C},\hat{A}}\mathcal{C}_{\hat{A},\hat{B}}Y. \tag{A.11}$$

The proof will be completed if we show that for all $1 \le f \le \beta$, $1 \le g \le \delta$, for all integer k such that $1 \le k \le f+g-1$, the derivative

$$\frac{d^{f+g-k-1}}{ds^{f+g-k-1}} \left\{ x^{(f)}(s)(T(s) - \hat{T}(s))y^{(g)}(s) \right\} \Big|_{s=z} = 0. \tag{A.12}$$

Let us first verify (A.12) for $k = 1$. First, straightforward calculation gives

$$\frac{d^{f+g-2}}{ds^{f+g-2}} \left\{ x^{(f)}(s)T(s)y^{(g)}(s) \right\} \Big|_{s=z}$$

$$= \sum_{k=0}^{f-1} \sum_{l=0}^{g-1} x^{[k]} C(zI - A)^{k+l-f-g+1} B y^{[l]} \tag{A.13}$$

$$= \sum_{k=0}^{f-1} \sum_{l=0}^{g-1} (x^{[k]} C(zI - A)^{k-f})(zI - A)((zI - A)^{l-g} B y^{[l]}) \tag{A.14}$$

$$= \left(X\mathcal{O}_{C,A}(zI - A)\mathcal{C}_{A,B}Y\right)_{f,g}. \tag{A.15}$$

From Lemmas 3.7 and 3.9,

$$(X\mathcal{O}_{C,A}(zI - A)\mathcal{C}_{A,B}Y) = (X\mathcal{O}_{\hat{C},\hat{A}}(zI - \hat{A})\mathcal{C}_{\hat{A},\hat{B}}Y). \tag{A.16}$$

This concludes the proof for the case $k = 1$. Now, we assume that for all $1 \le f \le \beta$ and $1 \le g \le \delta$, and for all $0 \le r \le \min(k, f+g-1)$,

$$\frac{d^{f+g-r-1}}{ds^{f+g-r-1}} \left\{ x^{(f)}(s)(T(s) - \hat{T}(s))y^{(g)}(s) \right\} \Big|_{s=z} = 0, \tag{A.17}$$

and we want to prove that (A.17) is still true for $r = \min(k+1, f+g-1)$. So, we choose $1 \le f \le \beta$ and $1 \le g \le \delta$ such that $f+g-1 \ge k+1$. We obtain the following equations:

$$\frac{d^{f+g-k-2}}{ds^{f+g-k-2}} \left\{ x^{(f)}(s)(T(s) - \hat{T}(s))y^{(g)}(s) \right\} \Big|_{s=z}$$

$$= \frac{d^{f-1+g-k-1}}{ds^{f-1+g-k-1}} \left\{ x^{(f-1)}(s)(T(s) - \hat{T}(s))y^{(g)}(s) \right\} \Big|_{s=z} \tag{A.18}$$

$$+ \frac{d^{f-1+g-k-1}}{ds^{f-1+g-k-1}} \left\{ (z-s)^{f-1} x^{[f-1]}(T(s) - \hat{T}(s))y^{(g)}(s) \right\} \Big|_{s=z}. \tag{A.19}$$

By the recursive argument,

$$\frac{d^{f-1+g-k-1}}{ds^{f-1+g-k-1}} \left\{ x^{(f-1)}(s)(T(s) - \hat{T}(s))y^{(g)}(s) \right\} \Big|_{s=z} = 0. \tag{A.20}$$

Moreover, we know from (A.3) that

$$(T(s) - \hat{T}(s))y^{(g)}(s) = O(z-s)^g. \tag{A.21}$$

This implies that

$$\frac{d^{f+g-k-2}}{ds^{f+g-k-2}} \left\{ x^{(f)}(s)(T(s) - \hat{T}(s))y^{(g)}(s) \right\} \Big|_{s=z} = 0. \tag{A.22}$$

The case at infinity can be treated in a similar way.

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, 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.

[4] J. A. Ball, I. Gohberg, and L. Rodman, *Interpolation of Rational Matrix Functions*, Birkhäuser Verlag, Basel, 1990.

[5] E. Chiprout and M. S. Nakhla, *Asymptotic Waveform Evaluation and Moment Matching for Interconnect Analysis*, Kluwer Internat. Ser. Eng. Comput. Sci., Kluwer, Boston, 1994.

[6] S. E. Cohn, *An introduction to estimation theory*, J. Meteor. Soc. Japan, 75 (1997), pp. 257–288.

[7] C. de Villemagne and R. E. Skelton, *Model reductions using a projection formulation*, Internat. J. Control, 46 (1987), pp. 2141–2169.

[8] K. Gallivan, E. Grimme, and P. Van Dooren, *Model reduction of large-scale systems, rational Krylov versus balancing techniques*, in Error Control and Adaptivity in Scientific Computing, Kluwer Acad. Publ., Dordrecht, The Netherlands, 1999, pp. 177–190.

[9] K. Gallivan, A. Vandendorpe, and P. Van Dooren, *Model reduction via truncation: an interpolation point of view*, Linear Algebra Appl., 375 (2003), pp. 115–134.

[10] K. Gallivan, A. Vandendorpe, and P. Van Dooren, *Sylvester equations and projection-based model reduction*, J. Comp. Appl. Math., 162 (2004), pp. 213–229.

[11] E. Grimme, *Krylov Projection Methods for Model Reduction*, Ph.D. thesis, Department of Electrical Engineering, University of Illinois at Urbana-Champaign, 1997.

[12] P. Holmes, J. L. Lumley, and G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge Monogr. Mech., Cambridge University Press, Cambridge, UK, 1996.

[13] T. Mukherjee, G. Fedder, D. Ramaswamy, and J. White, *Emerging simulation approaches for micromachined devices*, IEEE Trans. Comput. Aided Design of Integrated Circuits and Systems, 19 (2000), pp. 1572–1589.

[14] H. H. Rosenbrock, *State-Space and Multivariable Theory*, John Wiley & Sons, New York, 1970.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 2, pp. 350–361

PSEUDOSPECTRAL COMPONENTS AND THE DISTANCE TO UNCONTROLLABILITY*

J. V. BURKE†, A. S. LEWIS‡, AND M. L. OVERTON§

Abstract. We show that 2-norm pseudospectra of m-by-n matrices have no more than $2m(4m-1)$ connected components. Such bounds are pertinent for computing the distance to uncontrollability of a control system, since this distance is the minimum value of a function whose level sets are pseudospectra. We also discuss algorithms for computing this distance, including a trisection variant of Gu's recent algorithm, and we show how these may be used to locally maximize the distance to uncontrollability for a parameterized system.

Key words. pseudospectrum, robust control, distance to uncontrollability, connected components, trisection

AMS subject classifications. Primary, 15A18, 93B05; Secondary, 65F15

DOI. 10.1137/S0895479803433313

1. Introduction.

For matrices A and B of sizes p-by-p and p-by-q, respectively, consider the control system defined by

$$\dot{x} = Ax + Bu.$$

Here, $x \in \mathbf{R}^p$ is the state vector, and $u \in \mathbf{R}^q$ is the control vector (both depending on time). This system is *controllable* if, given any initial and final states $x(0)$ and $x(T)$, there exists a control function $u(\cdot)$ giving a trajectory $x(\cdot)$ with the given endpoints. In practice A and B are usually real.

Classical theory (see, for example, [ZDG96]) provides a simple characterization of controllability. The above system is controllable exactly when the matrix $[A - zI \;\; B]$ has full row rank for all scalars $z \in \mathbf{C}$.

Given any square matrix A, it is well known that the distance to the nearest singular matrix (measured in the usual operator 2-norm) is given by the smallest singular value $\sigma_{\min}(A)$ and that the conditioning of linear systems involving A depends on this quantity. Another important measure is the distance from A to instability, that is, the distance to the nearest matrix, possibly complex even if A is real, with an eigenvalue in the closed right half-plane. This distance plays a key role in robust stability analysis of the dynamical system $\dot{x} = Ax$.

The analogous question for controllability asks for the distance from the pair (A, B) to the nearest pair (A', B'), possibly complex even if (A, B) is real, corresponding to an uncontrollable system. A small distance to uncontrollability correlates with various difficulties for the control system, including numerical challenges for associated "pole placement" problems. A simple argument based on the singular value

*Received by the editors August 12, 2003; accepted for publication (in revised form) by B. T. Kågström April 8, 2004; published electronically November 17, 2004.
http://www.siam.org/journals/simax/26-2/43331.html

†Department of Mathematics, University of Washington, Seattle, WA 98195 (burke@math.washington.edu, www.math.washington.edu/~burke). The research of this author was supported in part by National Science Foundation grant DMS-9971852.

‡Department of Mathematics, Simon Fraser University, Burnaby, BC V5A 1S6, Canada (aslewis@sfu.ca, www.cecm.sfu.ca/~aslewis). The research of this author was supported in part by NSERC.

§Courant Institute of Mathematical Sciences, New York University, New York, NY 10012 (overton@cs.nyu.edu, www.cs.nyu.edu/~overton). The research of this author was supported in part by National Science Foundation grant CCR-0098145.

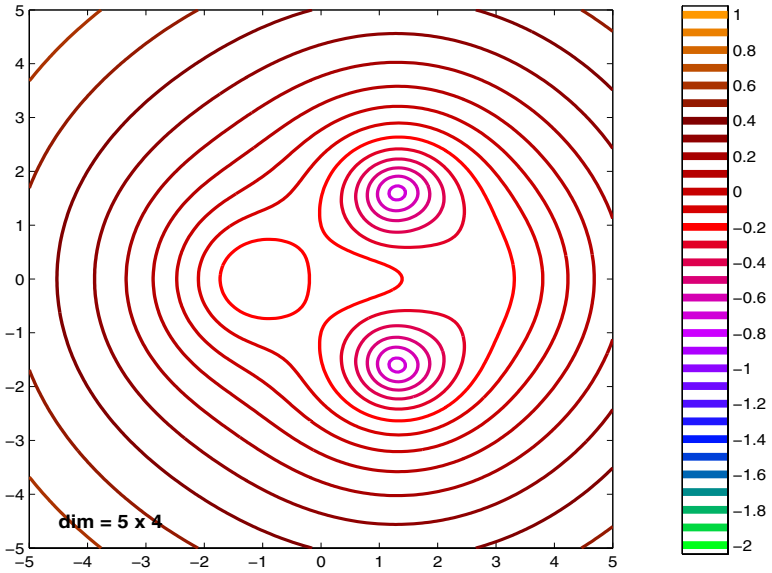


FIG. 1. *Pseudospectra for the controllable pair* (1.2) *with* $x_1 = x_2 = 1$.

decomposition [Eis84] shows that the distance to uncontrollability is given by

$$\min_{z \in \mathbf{C}} \sigma_{\min}[A - zI \;\; B], \tag{1.1}$$

a global optimization problem in two real variables. Here $\sigma_{\min}$ of a p-by-$p+q$ matrix C means the square root of the smallest eigenvalue of CC^*, a positive quantity when C has rank p.

The function to be minimized in the expression (1.1) has lower level sets of the form

$$\{z \in \mathbf{C} : \sigma_{\min}[A - zI \;\; B] \le \epsilon\}$$

for real $\epsilon > 0$. These sets, commonly called *pseudospectra*, have been well studied for square matrices, when the matrix B is empty; see the Pseudospectra Gateway [ET]. Pseudospectra are less well understood in the rectangular case, but references include [TT96, WT01, HT02, WT02, BEGM03]. Substantial insight is gained from examples, so consider the parameterized matrix pair

$$(A, B)(x_1, x_2) = \left(\begin{bmatrix} 1 & 1 & 2 & 3 \\ -1 & 1 & 4 & 5 \\ 0 & x_1 & 1 & 2 \\ x_2 & 0 & -2 & 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \right), \tag{1.2}$$

where x_1 and x_2 are real parameters. Figures 1 and 2 show pseudospectra[1] for, respectively, the controllable pair (1.2) when $x_1 = x_2 = 1$ and the uncontrollable pair (1.2) when $x_1 = x_2 = 0$ (the latter case being an example from [Gu00]).

The horizontal and vertical axes in the figures show the real and imaginary parts of z. The legends on the right sides of the figures show the contour heights (values

[1] All the figures in this paper were produced using T. Wright's software EigTool [Wri02].

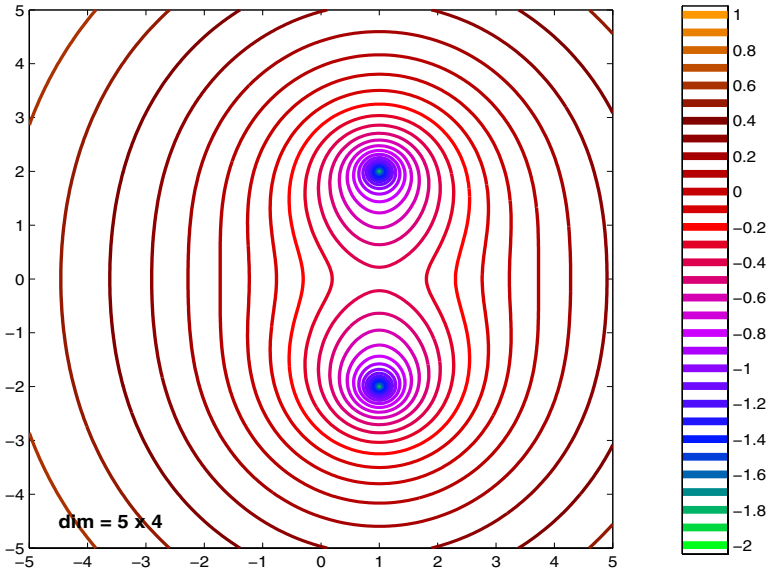


FIG. 2. *Pseudospectra for the uncontrollable pair* (1.2) *with* $x_1 = x_2 = 0$.

of ϵ) on a log 10 scale, with both plots using the same scale. In Figure 1, the "pseudospectral landscape" has three local minimizers and one can estimate that the global minimum value (by definition, the distance to uncontrollability) is about $10^{-0.7}$ (in fact, it is 0.1872). In Figure 2, there are only two local minimizers (forming a complex conjugate pair), and one can see that the contours drop to much lower values (in fact, it is easy to check that the minimum value of (1.1) is zero at the points $z = 1 \pm 2i$). In Figure 2 it is clear that some pseudospectra contain two connected components. In Figure 1, it is not clear, without a more detailed analysis, whether there are values of ϵ for which the pseudospectra have three connected components (in fact, there do exist such ϵ).

Our aim in this work is to find an upper bound on the number of connected components in the pseudospectra of rectangular matrices. We use a slightly more general setting than described above, defining

$$\Lambda = \{z \in \mathbf{C} : \sigma_{\min}(P + zQ) \leq \epsilon\} \tag{1.3}$$

for given matrices P and Q in the space $\mathbf{M}^{m,n}$ of m-by-n complex matrices (with $m \leq n$). In the case above we have $P = [A\ B]$ and $Q = [-I\ 0]$. Our goal is to find an upper bound on the number of components of the set Λ. Specifically, we show this number is no more than $2m(4m - 1)$. To our knowledge, this general bound is the best known, although it is certainly not tight. In particular, it is well known that pseudospectra of m-by-m matrices have no more than m components, since each component contains an eigenvalue [Tre97]. Furthermore, in the case of a single row ($m = 1$), it is easy to see that each nonempty pseudospectrum is simply a circular disk. We are not aware of an example of a pseudospectrum with more than m components.

We hope our analysis of pseudospectral components will shed light on the complexity of the problem of computing the distance to uncontrollability, for which we discuss algorithms in the second half of the paper. We begin by discussing a recent algorithm due to Gu [Gu00] for estimating the distance to uncontrollability within

a factor of two, and we show how a trisection variant can be used to obtain any prescribed accuracy. We then discuss an algorithm that combines repeated local optimization with Gu's algorithm and speculate that techniques similar to those used in analyzing the number of pseudospectral components might be used to bound the number of local optimization steps in this process.

Finally, with an effective algorithm in hand to evaluate the distance to uncontrollability (and, where defined, its gradient), we consider local maximization of the distance to uncontrollability for a smoothly varying parameterized pair (A, B) over a vector of free parameters. For the family (1.2), we find a locally maximizing pair with pseudospectra having four components.

2. Generic properties of singular values. To prove an upper bound on the number of components of the set Λ defined by (1.3), we first dispose of some trivial cases. Clearly we can suppose Q is nonzero, and hence Λ is compact. Furthermore, only the case $\epsilon \geq 0$ is interesting, as otherwise Λ is empty.

When $\epsilon = 0$, the set Λ either is the whole complex plane or consists of at most m points, as the following argument shows. Notice Λ is just the set of complex z for which $P + zQ$ has rank less than m. Assuming Λ is not the whole plane, we lose no generality in supposing that it does not contain zero or, in other words, that the matrix P has rank m. Partition the matrices P and Q as $[P_1 \;\; P_2]$ and $[Q_1 \;\; Q_2]$, respectively, where P_1 and Q_1 are m-by-m, and, again without loss of generality, P_1 is invertible. Since the function $\det(P_1 + zQ_1)$ is a polynomial of degree at most m, and is nonzero at zero, it has at most m zeros. But this set of zeros contains Λ, so the claim follows.

Henceforth we therefore assume $\epsilon > 0$. In the case $m = 1$, an easy calculation shows that Λ is either empty or a circular disk.

Our goal in this section is to show that for a fixed $\delta > 0$ and a "generic" matrix P, the singular value $\sigma_{\min}(P + zQ)$ is always either simple or less than δ. The proof is based on the following classical result in the space of m-by-m Hermitian matrices $\mathbf{H}^m$ (a real vector space of dimension m^2), concerning matrices X with a multiple smallest eigenvalue $\lambda_{\min}(X)$.

THEOREM 2.1 (von Neumann and Wigner [vNW29]). *For any integer $m > 1$, the algebraic set*

$$\hat{\mathbf{H}}^m = \{X \in \mathbf{H}^m : \lambda_{\min}(X) \text{ multiple}\}$$

has real codimension 3.

For example, the space $\mathbf{H}^2$ has dimension 4, and the set $\hat{\mathbf{H}}^2$ consists simply of real multiples of the identity matrix.

We also need an elementary supporting result.

PROPOSITION 2.2 (surjectivity). *A matrix $Y \in \mathbf{M}^{m,n}$ has full row rank if and only if the function $X \mapsto XY^* + YX^*$ maps $\mathbf{M}^{m,n}$ onto $\mathbf{H}^m$.*

Proof. Denote the given function by $\Phi : \mathbf{M}^{m,n} \to \mathbf{H}^m$. If Y has full row rank, then, with no loss of generality, $Y = [Y_0 \; Y_1]$, where the matrix Y_0 is invertible. Now, given any matrix $E \in \mathbf{H}^m$, we have $\Phi(\frac{1}{2}[EY_0^{-*} \; 0]) = E$, so Φ is indeed onto.

Conversely, suppose the map Φ is onto, and some $x \in \mathbf{C}^m$ satisfies $Y^*x = 0$. Choose a matrix $X \in \mathbf{H}^m$ satisfying $\Phi(X) = xx^*$. Then

$$\|x\|^4 = x^*(XY^* + YX^*)x = 0,$$

so $x = 0$, as required. ∎

We are now ready for the main result of this section.

THEOREM 2.3 (generic singular values). *For any $n \geq m > 1$ and any real $\delta > 0$, the real semialgebraic set*

$$\{Y \in \mathbf{M}^{m,n} : \sigma_{\min}(Y) \text{ is both multiple and at least } \delta\}$$

has real codimension 3.

Proof. Define a map $\Psi : \mathbf{M}^{m,n} \to \mathbf{H}^m$ by $\Psi(Y) = YY^*$. Notice that the given set, which we denote S, is defined locally by the inverse image $\Psi^{-1}(\hat{\mathbf{H}}^m)$. Furthermore, any $Y \in S$ has full row rank, and so Proposition 2.2 shows that the derivative $\nabla\Psi(Y)$ is onto. Since $\hat{\mathbf{H}}^m$ has codimension 3 by the result of von Neumann and Wigner (Theorem 1.1), so does S. ∎

COROLLARY 2.4. *For any $n \geq m > 1$, real $\delta > 0$, and matrix $Q \in \mathbf{M}^{m,n}$, the real semialgebraic set*

$$\{P \in \mathbf{M}^{m,n} : \exists z \in \mathbf{C} \text{ so } \sigma_{\min}(P + zQ) \text{ is both multiple and at least } \delta\}$$

has real codimension at least 1.

It follows from this last corollary that for a generic matrix P, the singular value $\sigma_{\min}(P + zQ)$ is always either simple or less than δ.

3. The generic case. Our bound on the number of components of pseudospectra relies on the following classical result [Mil64].

THEOREM 3.1 (Milnor). *For any polynomial $p : \mathbf{R}^2 \to \mathbf{R}$ of degree d, the zero set $p^{-1}(0)$ has no more than $d(2d-1)$ components.*

(In fact Milnor bounds the sum of the Betti numbers of $p^{-1}(0)$: the result above follows from the fact that the number of components is the zeroth Betti number.)

To apply Milnor's result, we need to relate the number of components of pseudospectra to their boundaries. We accomplish this with the following elementary result.

PROPOSITION 3.2 (components and boundaries). *Consider any continuous function $f : \mathbf{C} \to \mathbf{R}$. If the zero set $f^{-1}(0)$ is nonempty, then it has at least as many components as the level set $f^{-1}(-\infty, 0]$.*

Proof. Denote the zero set by E and the level set by L. It suffices to show that every component of L contains a component of E. If this is not the case, L has a component L_1 contained in the set $L' = f^{-1}((-\infty, 0))$. By continuity, L is closed and L' is open. Hence L_1, which is a component of both sets, must be both closed and open, and hence equal to the whole complex plane. Thus E must be empty, contrary to assumption. ∎

Using this technique in conjunction with Milnor's theorem, we can now prove our basic result.

THEOREM 3.3 (generic case). *Given any real ϵ and matrices $P, Q \in \mathbf{M}^{m,n}$ (where $m \leq n$), suppose there exists no complex z for which the singular value $\sigma_{\min}(P + zQ)$ is both multiple and equals ϵ. Then the set*

$$\Lambda = \{z \in \mathbf{C} : \sigma_{\min}(P + zQ) \leq \epsilon\}$$

has no more than $2m(4m-1)$ components.

Proof. The case $m = 1$ is elementary, so we suppose $m > 1$. For any matrix $A \in \mathbf{M}^{m,n}$, we write the singular values of A by multiplicity and in decreasing order: $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_m(A)$. In this notation, $\sigma_{\min}(A) = \sigma_m(A)$.

Consider the two disjoint open sets

$$\Gamma_< = \{z \in \mathbf{C} : (\sigma_m + \sigma_{m-1})(P + zQ) < 2\epsilon\},$$
$$\Gamma_> = \{z \in \mathbf{C} : (\sigma_m + \sigma_{m-1})(P + zQ) > 2\epsilon\}.$$

By assumption, the set

$$\Lambda' = \{z \in \mathbf{C} : \sigma_{\min}(P + zQ) = \epsilon\}$$

is contained in $\Gamma_>$, whereas the set

$$\Lambda'' = \bigcup_{j=1}^{m-1} \{z \in \mathbf{C} : \sigma_j(P + zQ) = \epsilon\}$$

is contained in $\Gamma_<$. Hence Λ' has no more components than the set

$$\begin{aligned}\Lambda' \cup \Lambda'' &= \bigcup_{j=1}^{m} \{z \in \mathbf{C} : \sigma_j(P + zQ) = \epsilon\} \\ &= \{z \in \mathbf{C} : \det((P + zQ)(P + zQ)^* - \epsilon^2 I) = 0\}.\end{aligned}$$

We can suppose that the matrix Q is nonzero and that the set Λ is nonempty. Applying Proposition 3.2 (components and boundaries) to the function $f(z) = \sigma_{\min}(P + zQ) - \epsilon$ shows that Λ has no more components than Λ', and hence no more than $\Lambda' \cup \Lambda''$.

The function $\phi : \mathbf{C}^2 \to \mathbf{C}$ defined by

$$\phi(x, y) = \det((P + (x + iy)Q)(P + (x + iy)Q)^* - \epsilon^2 I)$$

is clearly a polynomial of degree $2m$. Since Hermitian matrices have real determinants, $\phi(x, y)$ is real for all real x and y. Thus the restriction $\phi|_{\mathbf{R}^2}$ is a polynomial of degree $2m$ (whose coefficients we could identify by partial differentiation). The zero set of this polynomial is

$$\{(x, y) \in \mathbf{R}^2 : x + iy \in \Lambda' \cup \Lambda''\},$$

and our result now follows by applying Milnor's theorem (3.1). ∎

4. The general case. We extend our basic result, Theorem 3.3 (generic case), to the general case by a limiting argument. Recall that a sequence of subsets S_r of some Euclidean space *converges* to another set S if the following properties hold [RW98]:

(i) For any point $x \in S$, there exists a sequence of points $x_r \in S_r$ whose limit is x.

(ii) For any subsequence R of $\mathbf{N}$, the limit of any convergent sequence of points $x_r \in S_r$ $(r \in R)$ lies in S.

We first prove that, with this notion of convergence, the number of components of a compact set has a lower semicontinuity property.

PROPOSITION 4.1 (lower semicontinuity). *Consider a sequence of closed subsets of S_r of a Euclidean space converging to a compact set S. If S has a finite number of components, say k, then S_r has at least k components for all large r.*

Proof. We can suppose the set S is nonempty. Denote its components by S^j $(j = 1, 2, \ldots, k)$, and the closed and open unit balls by B and B°, respectively. Components of compact sets are compact, so for some real $\delta > 0$, the sets $S^j + \delta B$ $(j = 1, 2, \ldots, k)$ are disjoint. Choose real M so that $S + \delta B \subset MB$.

We first claim

$$S_r \subset (S + \delta B^\circ) \cup MB^c \quad \text{for all large } r.$$

Otherwise there would be a subsequence R of $\mathbf{N}$ and points

$$x_r \in S_r \cap (S + \delta B^\circ)^c \cap MB \quad (r \in R).$$

This bounded sequence has a cluster point in the closed set $(S + \delta B^\circ)^c$, contradicting the fact that the sets S_r converge to S.

Thus for all large r, the set S_r is contained in the disjoint union of open sets

$$MB^c \cup \bigcup_{j=1}^{k} (S^j + \delta B^\circ).$$

If the result fails, then the number of components of S_r is strictly less than k for all r in some subsequence R of $\mathbf{N}$. Hence for some index j and a further subsequence R' of R, we must have

$$S_r \cap (S^j + \delta B^\circ) = \emptyset \quad \text{for all } r \in R'.$$

But this contradicts the definition of convergence, since for any point $x \in S^j$ there exists a sequence of points $x_r \in S_r$ converging to x. ∎

Using this result, we can prove our main result.

THEOREM 4.2 (components of pseudospectra). *For any matrices $P, Q \in \mathbf{M}^{m,n}$ (where $m \leq n$) and any real ϵ, the set $\{z \in \mathbf{C} : \sigma_{\min}(P + zQ) \leq \epsilon\}$ has no more than $2m(4m-1)$ components.*

Proof. We can suppose that the given set, which we denote by Λ, is nonempty, that $\epsilon > 0$, and that Q is nonzero.

By Corollary 2.4, there exists a sequence of matrices $P_r \in \mathbf{M}^{m,n}$ satisfying the following two conditions:

(i) $\|P_r - P\| \leq 1/r$.

(ii) For no $z \in \mathbf{C}$ is $\sigma_{\min}(P_r + zQ)$ both multiple and equal to $\epsilon + 1/r$.

It follows by Theorem 3.3 (generic case) that the set

$$\Lambda_r = \left\{ z \in \mathbf{C} : \sigma_{\min}(P_r + zQ) \leq \epsilon + \frac{1}{r} \right\}$$

has no more than $2m(4m-1)$ components.

Using a well-known property of singular values, any point $z \in \Lambda$ satisfies

$$\sigma_{\min}(P_r + zQ) \leq \sigma_1(P_r - P) + \sigma_{\min}(P + zQ) \leq \frac{1}{r} + \epsilon,$$

so $\Lambda \subset \Lambda_r$ for all r. On the other hand, the continuity of $\sigma_{\min}$ shows that any cluster point of a sequence of points $z_r \in \Lambda_r$ must lie in Λ. Thus the compact sets Λ_r converge to the compact set Λ.

Finally, notice that Λ, being semialgebraic, has finitely many components. Hence we can apply Proposition 4.1 (lower semicontinuity) to deduce that, in fact, it has no more than $2m(4m-1)$ components, as required. ∎

5. Computing the distance to uncontrollability. Let $\tau(A, B)$ denote the distance to uncontrollability for a pair (A, B), defined by (1.1), where A is p-by-p and B is p-by-q. Thus the problem of computing $\tau(A, B)$ is that of minimizing $\sigma_{\min}[A - zI \;\; B]$ over the whole complex plane, a global minimization problem in two real variables.

It is interesting to compare the difficulty of this problem with that of two others: computing the distance to singularity and the distance to instability for the p-by-p matrix A alone. Let us assume that the computation of the minimum singular value function $\sigma_{\min}$ is an atomic operation. Computing the distance to singularity (distance to the nearest singular matrix) then requires one evaluation of $\sigma_{\min}$, while the distance to instability (distance to the nearest unstable matrix) may be computed by minimizing $\sigma_{\min}(A - zI)$ over the imaginary axis (equivalently, a global optimization problem in one real variable). Computation of the distance to instability, say $\beta(A)$, is a standard operation in control. The key observation is that checking whether $\beta(A)$ is less than a fixed number δ simply requires checking whether an associated Hamiltonian matrix has any imaginary eigenvalues. This leads immediately to a bisection algorithm [Bye88, BS90] that evaluates $\beta(A)$ to any prescribed accuracy in exact arithmetic, taking the computation of eigenvalues of $2p$-by-$2p$ Hamiltonian matrices as another atomic operation. Higher-order convergent algorithms are also well known [BB90]. In practice, it is important to compute the eigenvalues of the Hamiltonian matrices by a special algorithm that preserves Hamiltonian structure (such as in [Van84]) in order to avoid numerical blunders that incorrectly identify an eigenvalue as nonimaginary because of unnecessary rounding errors in its real part. The Hamiltonian imaginary eigenvalue test in these algorithms may be replaced by a linear matrix inequality (LMI) test (see, e.g., [BTN01]). This is computationally more expensive in practice, but offers the advantage of a complexity analysis that does not require assumption of eigenvalue and singular value computation as atomic operations.

By contrast, computing the distance to uncontrollability $\tau(A, B)$ seems to be a harder problem, and there are no standard methods in use, though there have been some recent theoretical advances. In 2000, Gu published an algorithm [Gu00] that estimates $\tau(A, B)$ within a factor of two. Gu's algorithm is based on a most ingenious test ("Gu's test," for brevity) that compares imaginary eigenvalues of matrix pencils involving Kronecker products that depend on A and B. Taking the computation of singular values and eigenvalues as atomic operations that can be performed in time cubic in the matrix dimension, and assuming that $q = O(p)$ (in practice, typically $q < p$), Gu's test requires $O(p^6)$ operations. No other polynomial-time algorithm for estimating $\tau(A, B)$ within a constant factor seems to be known; in particular, it does not seem to be known whether Gu's test could be replaced by an LMI-based test.

Gu's test may be summarized as follows. Given two numbers δ_1 and δ_2 (known as δ and $\delta - \eta/2$, respectively, in [Gu00]), with $\delta_1 > \delta_2 > 0$, Gu's test returns either the information that

$$\tau(A, B) \le \delta_1 \tag{5.1}$$

or the information that

$$\tau(A, B) > \delta_2. \tag{5.2}$$

At least one of these statements must be true; even if both are true, only one of the two statements is verified. As already noted, Gu's test involves comparing imaginary

eigenvalues of matrix pencils. We note for the record that both the terms $Q_{12} \otimes A$ and $I \otimes (A^* Q_{12})$ in the definition of $\mathcal{A}$ in [Gu00, p. 996] have incorrect signs.

Gu's estimation algorithm is then as follows.

ALGORITHM 5.1 (Gu's estimation algorithm).

1. *Set* $\delta_1 = \sigma_{\min}([A\ B])$, **done** = **false**.
2. *While not* **done**
   (a) *Set* $\delta_2 = \delta_1/2$.
   (b) *Perform Gu's test. If* (5.1) *is verified, set* $\delta_1 = \delta_2$*; if* (5.2) *is verified, set* **done** = **true**.

In exact arithmetic, this algorithm evaluates a nonzero $\tau(A, B)$ within a factor of two, but does not terminate if $\tau(A, B) = 0$.

It is tempting to try to evaluate $\tau(A, B)$ to higher precision by a bisection method. In order to make this work, one needs to set δ_1 and δ_2 sufficiently close to each other (η sufficiently small in the notation of [Gu00]) that (5.1) and (5.2) are almost mutually exclusive. Unfortunately, this leads to numerical difficulties; the necessary comparison of imaginary eigenvalues of the relevant pencils cannot be carried out with any confidence in the presence of rounding errors. However, a trisection variant works well, as follows.

ALGORITHM 5.2 (trisection variant of Gu's algorithm).

1. *Set* $L = 0$, $U = \sigma_{\min}([A\ B])$, **done** = **false**, **tol** *to a positive tolerance.*
2. *While not* **done**
   (a) *Set* $\delta_1 = L + 2(U - L)/3$ *and* $\delta_2 = L + (U - L)/3$.
   (b) *Perform Gu's test. If* (5.1) *is verified, set* $U = \delta_1$*; if* (5.2) *is verified, set* $L = \delta_2$.
   (c) *If* $U - L <$ **tol**, *set* **done** = **true**.

This trisection algorithm maintains upper and lower bounds U and L on $\tau(A, B)$, reducing the length of the interval $[L, U]$ by a factor of $2/3$ at each step of the iteration, and thereby computing $\tau(A, B)$ to any prescribed absolute accuracy in exact arithmetic in $O(p^6)$ operations. Furthermore, it is effective in practice even in the presence of rounding errors, running into numerical trouble only when $\tau(A, B)$ is determined at least to a few accurate digits.

An algorithm that runs much faster in practice, but without any complexity guarantee, is based on local optimization. This algorithm repeatedly performs local optimization of (1.1) using, for example, the well-known BFGS method. For controllable pairs, one expects the objective in (1.1) to be differentiable at minimizers, since the least singular value is being minimized, not maximized. As long as the least singular value at a local minimizer is simple and nonzero, the objective in (1.1) is continuously differentiable there. The BFGS algorithm requires the gradient of $\sigma_{\min}[A - zI\ \ B]$ with respect to the real and imaginary parts of z, which is given by

$$\begin{bmatrix} \operatorname{Re}\left(([I\ 0]u)^* v\right) \\ \operatorname{Im}\left(([I\ 0]u)^* v\right) \end{bmatrix},$$

where u and v are, respectively, the left and right singular vectors corresponding to $\sigma_{\min}[A - zI\ \ B]$. (One could use Newton's method instead of BFGS, as the corresponding 2-by-2 Hessian matrix is not hard to derive, but BFGS is so fast that Newton's method offers no significant advantage.) Once a local minimizer is obtained, Gu's test is used either to (i) verify global optimality within a tolerance or (ii) restart BFGS. A key point is that when Gu's test verifies (5.1), it also provides $\hat{z}$ for which $\sigma_{\min}[A - \hat{z}I\ \ B] = \delta_1$.

ALGORITHM 5.3 (BFGS/Gu hybrid).

1. *Set* $U = \sigma_{\min}([A\ B])$, $z = 0$, **done** = **false**, **tol** *to a positive tolerance.*
2. *While not* **done**
   (a) *Run BFGS starting at* z*, producing an approximate local minimizer* $\tilde{z}$*. Set* $\tilde{f} = \sigma_{\min}[A - \tilde{z}I\ \ B]$.
   (b) *Set* $\delta_1 = \tilde{f}(1 - 0.5\ \mathbf{tol})$ *and* $\delta_2 = \tilde{f}(1 - \mathbf{tol})$.
   (c) *Perform Gu's test. If* (5.1) *is verified, set* $U = \delta_1$ *and* $z = \hat{z}$*, where* $\sigma_{\min}[A - \hat{z}I\ \ B] = \delta_1$*; if* (5.2) *is verified, set* $L = \delta_2$ *and* **done** = **true**.

Although the objective function in (1.1) may have infinitely many local minimizers [GdH99], it has only finitely many locally minimal values (being semialgebraic). Assuming that an idealized BFGS algorithm always generates an exact local minimizer, in exact arithmetic the BFGS/Gu hybrid is guaranteed to terminate with an estimate of a nonzero $\tau(A, B)$ within any prescribed relative accuracy. A natural question is: how many local minimizers might be visited before a global optimizer is obtained? Unfortunately, our bound on the number of connected components does not immediately yield a bound on the number of locally minimal values. Nonetheless, we think that it might be possible to obtain a bound on the latter quantity by extending the techniques used to bound the former.

Our MATLAB implementations of the algorithms described in this section are freely available.[2]

6. Maximizing the distance to uncontrollability for a parameterized matrix pair. Finally, with two effective algorithms to evaluate $\tau(A, B)$ available, namely, the trisection variant of Gu's algorithm and the BFGS/Gu hybrid, we consider maximization of the distance to uncontrollability for a smoothly varying parameterized pair $(A, B)(x)$ over a vector of free parameters $x \in \mathbf{R}^k$. There are two reasons why this might be of interest. The first is that it may well be useful in applications to be able to find a matrix pair that is optimally far away from uncontrollability with respect to given free parameters. The second reason is that maximizing the distance to uncontrollability tends to produce pseudospectra with several isolated local minimizers whose minimal values are equal, and therefore is likely to produce pseudospectra with more components than would be found by randomly generating matrix pairs.

It is not difficult to see that the function $\tau(\cdot)$ is not differentiable on the space of (real or complex) matrix pairs; furthermore, it is easy to construct parameterized examples where the composite parameter-dependent function $\tau((A, B)(\cdot))$ is not differentiable at its maximizer. Such functions are not amenable to optimization by standard methods, such as BFGS, so we use a more specialized "gradient sampling" algorithm [BLO03]. This method exploits the fact that $\tau(\cdot)$ is differentiable almost everywhere, with gradient given by uv^*, where u and v are, respectively, the relevant left and right singular vectors for the matrix minimizing $\sigma_{\min}[A - zI\ \ B]$ over z, as long as the minimum singular value is simple and nonzero. We omit further details here and conclude by considering the example in (1.2). Running the gradient sampling algorithm to locally maximize the distance to uncontrollability over x_1 and x_2, we found an approximate local maximizer $\hat{x}$ given by $\hat{x}_1 = 1.9787$, $\hat{x}_2 = -1.8667$, with $\hat{\tau} = \tau((A, B)(\hat{x})) = 0.4897$. The corresponding pseudospectra are shown in Figure 3. Notice that the lowest points in this "pseudospectral landscape" are higher than those in Figure 1 and, furthermore, that four local minimizers have the same minimal value (namely, $\hat{\tau}$). Only two of the local minimizers occur in a complex conjugate pair; the

[2] http://www.cs.nyu.edu/faculty/overton/software

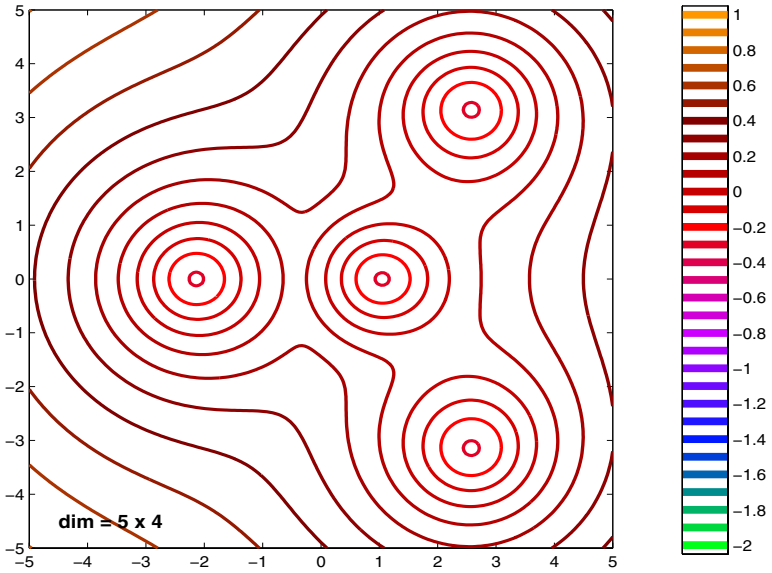


FIG. 3. *Pseudospectra for a local maximizer of* τ *over* (1.2).

other "ties" occur as a result of optimization over the parameters, with $\tau((A,B)(\cdot))$ not differentiable at its maximizer $\hat{x}$ as a result. Since there are four isolated local minimizers with minimal value $\hat{\tau}$, it follows that the pseudospectra have four components for ϵ in a range above $\hat{\tau}$. In this example, the row dimension p in fact equals four. Whether it is possible to produce pseudospectra with more than p components remains an open question.

Note added in proof. In fact, the bound $d(2d-1)$ in Milnor's result (Theorem 3.1) can be replaced by the sharp bound $(d^2-d+2)/2$ [BR90, Exercise 4.4.4], resulting in the improvement of our bound in Theorem 4.2 from $2m(4m-1)$ to $2m^2-m+1$. Whether a subquadratic bound holds is still an open question.

Acknowledgments. Many thanks to Nick Trefethen for posing the question of how to bound the number of pseudospectral components and for various helpful comments on this paper. Many thanks also to Ricky Pollack for pointing out Milnor's result.

REFERENCES

[BB90] S. BOYD AND V. BALAKRISHNAN, *A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its* L_∞*-norm*, Systems Control Lett., 15 (1990), pp. 1–7.

[BEGM03] B. BOUTRY, M. ELAD, G.H. GOLUB, AND P. MILANFAR, *The generalized eigenvalue problem for nonsquare pencils using a minimal perturbation approach*, SIAM J. Matrix Anal. Appl., to appear.

[BLO03] J.V. BURKE, A.S. LEWIS, AND M.L. OVERTON, *A robust gradient sampling algorithm for nonsmooth, nonconvex optimization*, SIAM J. Optim., to appear.

[BR90] R. BENEDETTI AND J.-J. RISLER, *Real Algebraic and Semi-algebraic Sets*, Hermann, Paris, 1990.

[BS90] N.A. BRUINSMA AND M. STEINBUCH, *A fast algorithm to compute the* $\mathbf{H}_\infty$*-norm of a transfer function matrix*, Systems Control Lett., 14 (1990), pp. 287–293.

[BTN01] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, MPS/SIAM Ser. Optim. 2, SIAM, Philadelphia, 2001.

[Bye88] R. Byers, *A bisection method for measuring the distance of a stable matrix to the unstable matrices*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 875–881.

[Eis84] R. Eising, *Between controllable and uncontrollable*, Systems Control Lett., 4 (1984), pp. 263–264.

[ET] M. Embree and L.N. Trefethen, *Pseudospectra Gateway*, http://web.comlab.ox.ac.uk/projects/pseudospectra/.

[GdH99] J.-M. Gracia and I. de Hoyos, *Nearest pairs with more nonconstant invariant factors and pseudospectra*, Linear Algebra Appl., 298 (1999), pp. 143–158.

[Gu00] M. Gu, *New methods for estimating the distance to uncontrollability*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 989–1003.

[HT02] N.J. Higham and F. Tisseur, *More on pseudospectra for polynomial eigenvalue problems and applications in control theory*, Linear Algebra Appl., 351/352 (2002), pp. 435–453.

[Mil64] J.W. Milnor, *On the Betti numbers of real varieties*, Proc. Amer. Math. Soc., 15 (1964), pp. 275–280.

[RW98] R.T. Rockafellar and R.J.-B. Wets, *Variational Analysis*, Springer, Berlin, 1998.

[Tre97] L.N. Trefethen, *Pseudospectra of linear operators*, SIAM Rev., 39 (1997), pp. 383–406.

[TT96] K.-C. Toh and L.N. Trefethen, *Calculation of pseudospectra by the Arnoldi iteration*, SIAM J. Sci. Comput., 17 (1996), pp. 1–15.

[Van84] C. Van Loan, *A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix*, Linear Algebra Appl., 61 (1984), pp. 233–251.

[vNW29] J. von Neumann and E. Wigner, *Über das Verhalten von Eigenwerten bei adiabatischen Prozessen*, Physik. Zeitschr., 30 (1929), pp. 467–470.

[Wri02] T.G. Wright, *EigTool: A Graphical Tool for Nonsymmetric Eigenproblems*, 2002; available online from http://web.comlab.ox.ac.uk/pseudospectra/eigtool/.

[WT01] T.G. Wright and L.N. Trefethen, *Large-scale computation of pseudospectra using ARPACK and eigs*, SIAM J. Sci. Comput., 23 (2001), pp. 591–605.

[WT02] T.G. Wright and L.N. Trefethen, *Pseudospectra of rectangular matrices*, IMA J. Numer. Anal., 22 (2002), pp. 501–519.

[ZDG96] 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. 26, No. 2, pp. 362–376

THE MATRIX SQUARE ROOT FROM A NEW FUNCTIONAL PERSPECTIVE: THEORETICAL RESULTS AND COMPUTATIONAL ISSUES*

BEATRICE MEINI†

Abstract. We give a new characterization of the matrix square root and a new algorithm for its computation. We show how the matrix square root is related to the constant block coefficient of the inverse of a suitable matrix Laurent polynomial. This fact, besides giving a new interpretation of the matrix square root, allows one to design an efficient algorithm for its computation. The algorithm, which is mathematically equivalent to Newton's method, is quadratically convergent and numerically insensitive to the ill-conditioning of the original matrix and works also in the special case where the original matrix is singular and has a square root.

Key words. matrix square root, matrix Laurent polynomial, cyclic reduction, Newton's method

AMS subject classifications. 15A24, 65F10, 65F30

DOI. 10.1137/S0895479803426656

1. Introduction. Let us denote by $\lambda(H)$ and $\rho(H)$ the set of the eigenvalues and the spectral radius of the square matrix H, respectively, by $\mathcal{D}$ the open unit disk in the complex plane, and by $\mathbb{C}^+$ ($\mathbb{C}^-$) the open right (left) half complex plane.

Let $A \in \mathbb{C}^{n\times n}$ be a matrix having no nonpositive real eigenvalues. Under this assumption, the quadratic matrix equation

$$X^2 - A = 0 \tag{1.1}$$

has a unique solution such that $\lambda(X) \subset \mathbb{C}^+$ (see [6, 10]). We will denote by $A^{1/2}$ that solution, which is usually called the principal square root of A.

The matrix square root has several properties and is closely related to the matrix sign function. A well-known fact is that [11]

$$\operatorname{sign}(A) = A(A^2)^{-1/2}.$$

Based on this property, in [12] Higham has shown that the matrix square root can be characterized in terms of the sign of a suitable 2×2 block matrix:

$$\operatorname{sign}\left(\begin{bmatrix} 0 & A \\ I & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & A^{1/2} \\ A^{-1/2} & 0 \end{bmatrix}. \tag{1.2}$$

This is a fundamental relation, since any method for computing the matrix sign can be used to compute the matrix square root. Indeed, Newton's iteration applied directly to the matrix equation (1.1) suffers from instability problems (see [17, 9]) while methods which are based on the computation of the matrix sign of the matrix in (1.2) are generally more stable [12].

In this paper we derive a new characterization of the matrix square root. More specifically we prove the following result.

*Received by the editors April 24, 2003; accepted for publication (in revised form) by N. J. Higham December 22, 2003; published electronically November 17, 2004.
http://www.siam.org/journals/simax/26-2/42665.html
†Dipartimento di Matematica, Università di Pisa, 56127 Pisa, Italy (meini@dm.unipi.it).

THEOREM 1.1. *Let $A \in \mathbb{C}^{n\times n}$ be a matrix having no nonpositive real eigenvalues. Then the matrix Laurent polynomial*

$$R(z) = (I - A)z^{-1} + 2(I + A) + (I - A)z \tag{1.3}$$

is invertible for any $z \in \mathbb{C}$ such that $r < |z| < 1/r$, where

$$r = \rho((A^{1/2} - I)(A^{1/2} + I)^{-1}) < 1.$$

Moreover, $H(z) = R(z)^{-1} = H_0 + \sum_{i=1}^{\infty} H_i(z^i + z^{-i})$ is such that

$$H_0 = \frac{1}{4}A^{-1/2}.$$

The theorem states that the block constant coefficient of the inverse of $R(z)$ fully defines the matrix square root. This is a new characterization in terms of matrix functions.

At first glance this property may appear to be of purely theoretical interest, since how to compute the constant coefficient of $H(z)$ is not immediately clear. However, we show that we can apply the cyclic reduction algorithm, which is quadratically convergent, has a low computational cost per iteration, and enjoys good numerical stability properties. In particular, the cyclic reduction algorithm, in contrast to other methods, works fine also in the critical case, where A is close to a singular matrix. If A is singular, convergence still occurs provided that the zero eigenvalues are semisimple, i.e., the Jordan blocks corresponding to 0 have size 1. In this case the convergence is linear. Unlike other methods for computing $A^{1/2}$, we do not apply scaling techniques in order to speed up the convergence; in fact, for particular problems our algorithm can be less efficient than other methods which use scaling. On this subject we refer the reader to the paper [14], where some scaling strategies for cyclic reduction are analyzed and the slow convergence is overcome.

The paper is organized as follows. In section 2 we prove Theorem 1.1. In section 3 we propose the algorithm, we prove its convergence properties, and we show some specific features of the algorithm for the singular case, the symmetric positive case, the case of M-matrices and the skew-Hamiltonian case. In section 4 we report the numerical results and the comparisons with the available methods. Finally, in section 5 we draw conclusions.

2. Proof of Theorem 1.1. Let us introduce the Cayley transformation

$$w : \mathbb{C} \setminus \{-1\} \to \mathbb{C}, \quad w(x) = \frac{x - 1}{x + 1},$$

which maps the right half plane $\mathbb{C}^+$ into the open unit disk $\mathcal{D}$. Since $\lambda(A^{1/2}) \subset \mathbb{C}^+$, the matrix

$$W = (A^{1/2} - I)(A^{1/2} + I)^{-1} \tag{2.1}$$

is well defined and is such that $\lambda(W) \subset \mathcal{D}$. Moreover, it is easy to verify that

$$A^{1/2} = (I + W)(I - W)^{-1}. \tag{2.2}$$

By substituting X in (1.1) with the above expression for $A^{1/2}$, since all the matrices commute, we obtain that W solves the matrix equation

$$(I - A) + 2(I + A)Y + (I - A)Y^2 = 0. \tag{2.3}$$

Moreover, from the uniqueness of $A^{1/2}$, W is the unique solution such that $\rho(W) < 1$.

We associate with the matrix equation (2.3) the matrix Laurent polynomial $R(z)$ of (1.3). Since W solves the matrix equation (2.3) and since $R(z)$ is a symmetric matrix Laurent polynomial it follows that $R(z)$ can be factorized as

$$R(z) = G(z)G(z^{-1}), \tag{2.4}$$

where

$$G(z) = z(A^{1/2} + I) - (A^{1/2} - I)$$

(this latter property can be also verified by direct inspection). In particular, $G(z)$ is invertible for any $z \in \mathbb{C}$ such that $\rho(W) < |z|$, and we may write

$$G(z)^{-1} = z^{-1}(A^{1/2} + I)^{-1}(I - z^{-1}W)^{-1} = z^{-1}(A^{1/2} + I)^{-1} \sum_{i=0}^{\infty} z^{-i}W^i.$$

Hence, from the factorization (2.4), we obtain

$$H(z) = R(z)^{-1} = G(z^{-1})^{-1}G(z)^{-1} = (A^{1/2} + I)^{-2} \sum_{i=0}^{\infty} z^i W^i \sum_{i=0}^{\infty} z^{-i}W^i,$$

which is convergent for any $z \in \mathbb{C}$ such that $\rho(W) < |z| < 1/\rho(W)$. The constant matrix coefficient of $H(z)$ is

$$H_0 = (A^{1/2} + I)^{-2} \sum_{i=0}^{\infty} W^{2i} = (A^{1/2} + I)^{-2}(I - W^2)^{-1} = \frac{1}{4}A^{-1/2}.$$

3. The algorithm. According to the results of [1], the cyclic reduction algorithm [4] applied to the bi-infinite block tridiagonal block Toeplitz matrix

$$\mathcal{T}_0 = \begin{bmatrix} \ddots & \ddots & & 0 \\ \ddots & 2(I+A) & (I-A) & \\ & (I-A) & 2(I+A) & \ddots \\ 0 & & \ddots & \ddots \end{bmatrix}$$

generates a sequence $\{\mathcal{T}_k\}_{k\geq 0}$ of bi-infinite block tridiagonal block Toeplitz matrices which quadratically converge to the block diagonal matrix having H_0^{-1} on the main diagonal, where H_0 is the constant term of the inverse of the matrix Laurent polynomial (1.3).

In this particular case, the cyclic reduction algorithm consists of generating the two sequences of matrices $\{Y_k\}_k$, $\{Z_k\}_k$ in the following way:

$$\begin{aligned} Z_0 &= 2(I + A), \quad Y_0 = I - A, \\ Y_{k+1} &= -Y_k Z_k^{-1} Y_k, \\ Z_{k+1} &= Z_k - 2Y_k Z_k^{-1} Y_k, \quad k = 0, 1, \ldots. \end{aligned} \tag{3.1}$$

At the kth step, Z_k is the matrix defining the main block diagonal of $\mathcal{T}_k$ and Y_k is the matrix defining the lower and the upper diagonal blocks of $\mathcal{T}_k$. Observe that, since both Y_k and Z_k are rational functions in A, Y_k and Z_k commute.

We now show some properties of the sequences $\{Y_k\}_k$, $\{Z_k\}_k$, from which we can also directly derive that the sequence $\{Z_k\}_k$ converges quadratically to $4A^{1/2}$.

PROPOSITION 3.1. *Let $A \in \mathbb{C}^{n\times n}$ be a matrix having no nonpositive real eigenvalues. Then the matrices Z_k generated by* (3.1) *are nonsingular for all $k \geq 0$ and*

$$Z_k + 2Y_k W^{2^k} = 4A^{1/2}, \qquad k = 0, 1, \ldots, \tag{3.2}$$

where W is defined in (2.1).

Proof. From the properties of cyclic reduction applied to matrix equations [2], the following identity holds:

$$Y_k W^{2^{k+1}} + Z_k W^{2^k} + Y_k = 0 \tag{3.3}$$

for $k = 0, 1, 2, \ldots$.

Let us prove by induction on k that Z_k is nonsingular. Z_0 is nonsingular since A cannot have an eigenvalue equal to -1. Let us assume the thesis true for a fixed $k \geq 0$ and let us prove it for $k+1$. From (3.3) we have

$$Z_k^{-1} Y_k = -W^{2^k}(I + W^{2\cdot 2^k})^{-1}. \tag{3.4}$$

Hence, from (3.1),

$$Z_{k+1} = Z_k \left(I - 2(Z_k^{-1}Y_k)^2\right) = Z_k \left(I - 2\left(W^{2^k}(I + W^{2\cdot 2^k})^{-1}\right)^2\right). \tag{3.5}$$

Since Z_k is nonsingular by inductive assumption, it is sufficient to show that $I - 2(W^{2^k}(I + W^{2\cdot 2^k})^{-1})^2$ is nonsingular. Any eigenvalue of this matrix has the form $\mu = 1 - 2\lambda^{2^{k+1}}(1 + \lambda^{2\cdot 2^k})^{-2}$, where λ is an eigenvalue of W. Since $\lambda(W) \subset \mathcal{D}$, μ cannot be zero.

Now from (3.3) it is a simple matter to verify by induction that

$$Z_k + 2Y_k W^{2^k} = Z_0 + 2Y_0 W$$

for any k. If we replace Z_0, Y_0, and W with their expression in terms of A and $A^{1/2}$ we obtain that

$$\begin{aligned} Z_0 + 2Y_0 W &= 2(I + A) + 2(I - A)(A^{1/2} - I)(I + A^{1/2})^{-1} \\ &= 2((I + A)(I + A^{1/2}) + (I - A)(A^{1/2} - I))(I + A^{1/2})^{-1} \\ &= 4A^{1/2}(I + A^{1/2})(I + A^{1/2})^{-1} = 4A^{1/2}. \end{aligned}$$

□

From the above proposition and from the spectral properties of W, we derive the following convergence results, which describe the quadratic convergence of our algorithm.

THEOREM 3.2. *Let $A \in \mathbb{C}^{n\times n}$ be a matrix having no nonpositive real eigenvalues. Then the sequences $\{Y_k\}_k$ and $\{Z_k\}_k$ of* (3.1) *are convergent and*

$$||Y_k|| = O(\sigma^{2^k}), \tag{3.6}$$

$$||Z_k - 4A^{1/2}|| = O(\sigma^{2\cdot 2^k}) \tag{3.7}$$

for any real number σ, $\rho(W) < \sigma < 1$, and for any matrix norm $||\cdot||$.

Proof. Let σ be any real number such that $\rho(W) < \sigma < 1$, and let $||\cdot||_\sigma$ be an induced matrix norm such that $||W||_\sigma \le \sigma$. From (3.4) we obtain that $||Z_k^{-1}Y_k||_\sigma \le \gamma\sigma^{2^k}$ for a suitable $\gamma > 0$. Therefore, from the equivalence of matrix norms, we obtain that

$$||Z_k^{-1}Y_k|| = O(\sigma^{2^k}) \tag{3.8}$$

for any matrix norm. Thus, from (3.5), it follows that the sequences $\{Z_k\}_k$ and $\{Z_k^{-1}\}_k$ have norm bounded above by a constant. Hence from (3.8) we deduce (3.6). Finally, from (3.2) and (3.6) we obtain (3.7). ∎

The resulting algorithm for the computation of $A^{1/2}$ consists of computing the sequences $\{Y_k\}_k$ and $\{Z_k\}_k$ of (3.1), until $||Y_k|| < \epsilon$, for a fixed error bound ϵ and for a chosen matrix norm $||\cdot||$, and then approximating $A^{1/2}$ by $Z_k/4$. It is worth noting that each step of our algorithm requires only one matrix inversion and 2 matrix multiplications.

It is also interesting to point out that the sequence $\{Z_k\}_k$ generated by our algorithm coincides, up to within the multiplicative constant 4, with the sequence generated by applying Newton's method to the matrix equation (1.1). More specifically, by denoting with $\{X_k\}_k$ the sequence generated by Newton's algorithm, i.e., $X_0 = A$, $X_{k+1} = (X_k + AX_k^{-1})/2$, $k \ge 0$, then we can easily verify by induction on k that

$$Y_k = (A - X_k^2)X_k^{-1},\ Z_k = 4X_{k+1},\ k = 0, 1, \ldots. \tag{3.9}$$

Observe that Y_k represents a "normalized" residual matrix of the approximation X_k. Thus we have a different way to compute the sequence generated by Newton's method, which is a little bit more expensive, but which, unlike the latter method, has good stability properties, as we will show by numerical experiments in section 4. This analogy with Newton's method is shared also by the Denman–Beavers algorithm [5], which is generally more stable than Newton's method [9]. Some different, but mathematically equivalent, ways to relate cyclic reduction and Newton's method are analyzed in [14]. Such different formulations of (3.9) allow one to perform a stability analysis of cyclic reduction and to derive some scaling strategies.

The singular case. Suppose that $A \in \mathbb{C}^{n\times n}$ is a singular matrix having no negative real eigenvalues, and suppose that the null eigenvalues are semisimple, i.e., their Jordan blocks have size 1. This hypothesis on the null eigenvalues is a necessary and sufficient condition for a singular complex matrix A to have a square root which is a function of A (see Theorem 6.4.12 in [13]).

We show that also in this case Z_k is nonsingular for any $k \ge 0$ and that $||Y_k|| = O(2^{-k})$, $||Z_k - 4A^{1/2}|| = O(2^{-k})$. Hence our algorithm can still be applied and converges linearly with rate 1/2. Note, however, that the sequence $\{Z_k\}_k$ converges to a singular matrix.

Under our assumption, if p is the number of eigenvalues equal to zero, the Jordan canonical form of A has the structure

$$J = PAP^{-1} = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & H \end{array}\right], \tag{3.10}$$

where H is a nonsingular block diagonal matrix of size $(n-p)\times(n-p)$. In particular

$$J^{1/2} = PA^{1/2}P^{-1} = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & H^{1/2} \end{array}\right].$$

Let us denote $\hat{W} = PWP^{-1}$, $\hat{Z}_k = PZ_kP^{-1}$, $\hat{Y}_k = PY_kP^{-1}$, $k \geq 0$. Then it is immediate to verify that $\hat{Y}_{k+1} = -\hat{Y}_k\hat{Z}_k^{-1}\hat{Y}_k$, $\hat{Z}_{k+1} = \hat{Z}_k - 2\hat{Y}_k\hat{Z}_k^{-1}\hat{Y}_k$, $k \geq 0$, and that $\hat{Z}_k + 2\hat{Y}_k\hat{W}^{2^k} = 4J^{1/2}$, $k \geq 0$. Moreover, $\hat{Y}_k$ and $\hat{Z}_k$, $k \geq 0$, have the structure

$$\hat{Y}_k = \left[\begin{array}{c|c} R_k^{(1)} & 0 \\ \hline 0 & R_k^{(2)} \end{array}\right], \quad \hat{Z}_k = \left[\begin{array}{c|c} S_k^{(1)} & 0 \\ \hline 0 & S_k^{(2)} \end{array}\right], \quad k \geq 0,$$

where $R_k^{(1)} = r_kI$, $S_k^{(1)} = s_kI$ are $p \times p$ diagonal matrices with equal diagonal entries. By following the same arguments used in section 3 we can show that $S_k^{(2)}$ is nonsingular for any k and that $||R_k^{(2)}|| = O(\sigma^{2^k})$, $||S_k^{(2)} - 4H^{1/2}|| = O(\sigma^{2\cdot 2^k})$, where σ is any real number $0 < \sigma < r$, and $r = \rho((H^{1/2} - I)(H^{1/2} + I)^{-1}) < 1$. Moreover, we have that

$$\begin{aligned} &r_0 = 1, \ s_0 = 2, \\ &r_{k+1} = -r_k^2/s_k, \ s_{k+1} = s_k - 2r_k^2/s_k, \quad k = 0, 1, \ldots. \end{aligned}$$

Hence $r_k = -2^{-k}$, $s_k = 2^{-k+1}$, $k = 1, 2, \ldots$, from which the nonsingularity of $\{Z_k\}_k$ and the convergence properties of $\{Z_k\}_k$ and $\{Y_k\}_k$ follow.

In the singular case we may also show that the residual matrix $\Gamma_k = A - (Z_k/4)^2$ and the error matrix $E_k = A^{1/2} - Z_k/4$, for $k \geq 0$, are such that

$$||\Gamma_k|| = O(||E_k||^2) \tag{3.11}$$

for any matrix norm. A similar property was observed by Guo and Laub in [7] for Newton's method applied to a particular algebraic Riccati equation.

In order to prove (3.11), define the matrix norm $||\cdot||_P$ as $||V||_P = ||PVP^{-1}||_\infty$, where V is an $n \times n$ matrix and P is the matrix of (3.10) such that $J = PAP^{-1}$. We show that, for this particular norm, if k is sufficiently large, then $||\Gamma_k||_P = ||E_k||_P^2$. Therefore (3.11) follows from the equivalence of matrix norms. Observe that

$$\begin{aligned} P\Gamma_kP^{-1} &= \left[\begin{array}{c|c} -(s_k/4)^2I & 0 \\ \hline 0 & H - (S_k^{(2)}/4)^2 \end{array}\right], \\ PE_kP^{-1} &= \left[\begin{array}{c|c} -(s_k/4)I & 0 \\ \hline 0 & H^{1/2} - S_k^{(2)}/4 \end{array}\right], \quad k \geq 0. \end{aligned}$$

Since the sequence $\{S_k^{(2)}/4\}_k$ converges quadratically to $H^{1/2}$, for k sufficiently large one has $||\Gamma_k||_P = ||-(s_k/4)^2I||_\infty = (s_k/4)^2$ and $||E_k||_P = ||-(s_k/4)I||_\infty = s_k/4$; therefore $||\Gamma_k||_P = ||E_k||_P^2$.

The symmetric positive definite case. In the case where A is real symmetric, then the matrices Y_k and Z_k are obviously symmetric for any k. If in addition A is positive definite (we will write $A > 0$), then it is possible to show that $-Y_k$ and Z_k are positive definite, and to give a bound on the spectral condition number of Z_k.

THEOREM 3.3. *If A is symmetric positive definite, then the matrices $-Y_k$, $k \geq 1$, and Z_k, $k \geq 0$, generated by* (3.1)*, are positive definite, and*

$$\kappa_2(Z_k) = \frac{\max\lambda(Z_k)}{\min\lambda(Z_k)} \leq \frac{1 + \max\lambda(A)}{2\sqrt{\min\lambda(A)}}, \quad k = 0, 1, \ldots.$$

Proof. Since W and Z_k are rational functions in $A^{1/2}$, from (3.5) it follows that the eigenvalues of Z_{k+1} are the eigenvalues of Z_k times the eigenvalues of the matrix

$H_k = I - 2(W^{2^k}(I + W^{2\cdot 2^k})^{-1})^2$; since the eigenvalues of W belong to the interval $(-1, 1)$, we deduce that the eigenvalues of H_k belong to the interval $(0, 1)$. Thus $Z_{k+1} > 0$ if $Z_k > 0$; since $Z_0 > 0$, we conclude that Z_k is positive definite for any k. Hence $-Y_k$, for $k \geq 1$, is also positive definite, and $Z_k - Z_{k+1}$ is positive definite for any $k \geq 0$; in particular, since $\lim_k Z_k = 4A^{1/2}$ and $Z_0 = 2(I + A)$, also $Z_k - 4A^{1/2}$ and $2(I + A) - Z_k$ are positive definite, and thus the eigenvalues of Z_k belong to the interval $[\mu_1, \mu_2]$, where $\mu_1 = 4 \min \lambda(A^{1/2})$, $\mu_2 = 2(1 + \max \lambda(A))$. □

From the above theorem it follows that the matrices Z_k, $k \geq 0$, which must be inverted at each step, can be better conditioned than A. In particular, assuming that $\max \lambda(A) = 1$ and $\min \lambda(A) < 1$, we always have

$$\kappa_2(Z_k) \leq \frac{1 + \max \lambda(A)}{2\sqrt{\min \lambda(A)}} = \frac{1}{\sqrt{\min \lambda(A)}} < \frac{1}{\min \lambda(A)} = \kappa_2(A).$$

The case of M-matrices. Also in the important case where A is an M-matrix, we can show some interesting properties of our algorithm. Up to within a scaling of A by its maximum diagonal entry, A may be written as $A = I - B$, where B is a nonnegative matrix such that $\rho(B) < 1$. Thus $Y_0 = I - A = B$ is nonnegative, and $Z_0 = 2(I + A) = 4I - 2B$ is an M-matrix. Moreover, observe that the matrices Y_k, Z_k, $k \geq 1$, generated by (3.1) do not change if we replace Y_0 with $-Y_0$; thus we may suppose $Y_0 = A - I = -B$. It is immediate to verify that, if $Z_0 = 4I - 2B$ and $Y_0 = -B$, then the $k \times k$ block matrix

$$H_k = \begin{bmatrix} Z_0 & Y_0 & & 0 \\ Y_0 & Z_0 & \ddots & \\ & \ddots & \ddots & Y_0 \\ 0 & & Y_0 & Z_0 \end{bmatrix}$$

is a nonsingular M-matrix for any $k \geq 1$. If we apply one step of block cyclic reduction to H_k, where $k = 2^q + 1$, $q \geq 1$, we obtain the permuted block LU factorization

$$\Pi H_k \Pi^T = \left[\begin{array}{cccc|ccc} Z_0 & & & 0 & Y_0 & & 0 \\ & \ddots & & & Y_0 & \ddots & \\ & & \ddots & & & \ddots & Y_0 \\ 0 & & & Z_0 & 0 & & Y_0 \\ \hline Y_0 & Y_0 & & 0 & Z_0 & & 0 \\ & \ddots & \ddots & & & \ddots & \\ 0 & & Y_0 & Y_0 & 0 & & Z_0 \end{array}\right] = \left[\begin{array}{c|c} V_1 & V_2 \\ \hline V_3 & V_4 \end{array}\right] = LU,$$

$$L = \left[\begin{array}{c|c} I & 0 \\ \hline V_3 V_1^{-1} & I \end{array}\right], \; U = \left[\begin{array}{c|c} V_1 & V_2 \\ \hline 0 & S \end{array}\right],$$

where Π is the block even-odd permutation matrix and where the Schur complement $S = V_4 - V_3 V_1^{-1} V_2$ is the $(2^{q-1} + 1) \times (2^{q-1} + 1)$ block matrix

$$S = \begin{bmatrix} Z_1 & Y_1 & & 0 \\ Y_1 & Z_1 & \ddots & \\ & \ddots & \ddots & Y_1 \\ 0 & & Y_1 & Z_1 \end{bmatrix}.$$

Since $\Pi H_k \Pi^T$ is a nonsingular M-matrix, for the properties of Schur complements, it follows that S is a nonsingular M-matrix; in particular Z_1 is an M-matrix, and Y_1 has nonpositive entries. We can prove by induction on k that Z_k is a nonsingular M-matrix, and Y_k has nonpositive entries for all $k \geq 0$. In particular, our algorithm has the same stability properties as block Gaussian elimination applied to M-matrices.

The skew-Hamiltonian case. Let us now consider the case where A is real skew-Hamiltonian, i.e., $n = 2m$ and A has the structure

$$A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_1^T \end{bmatrix},$$

where A_1, A_2, A_3 are $m \times m$ matrices, and A_2, A_3 are skew-symmetric. This situation occurs in the numerical solution of algebraic Riccati equations that is ultimately reduced to the computation of the square root of a matrix [20, 18].

Since A is skew-Hamiltonian, there exists an orthogonal matrix Q such that

$$B = Q^T A Q = \begin{bmatrix} B_1 & B_2 \\ 0 & B_1^T \end{bmatrix}, \tag{3.12}$$

where B_1 is upper Hessenberg and B_2 is skew-symmetric. Moreover, the reduced matrix $B = Q^T A Q$ can be computed in a numerically stable way by means of the Van Loan algorithm [19], and $A^{1/2} = Q B^{1/2} Q^T$. Indeed, the methods proposed in [20, 18] for computing the square root of a skew-Hamiltonian matrix A consist first of reducing A into the form (3.12), and then of approximating the square root of the reduced matrix B by exploiting the structure of B. Also our algorithm can be adapted for the computation of $B^{1/2}$, thus generating matrices which keep the structure of B, with a substantial reduction of the computational cost with respect to the direct application of the algorithm to the skew-Hamiltonian matrix A. Indeed, it is a simple matter to verify that the matrices $\{Y_k\}_k$ and $\{Z_k\}_k$, obtained by starting with $Y_0 = I - B$, $Z_0 = 2(I + B)$, have the structure:

$$Y_k = \begin{bmatrix} Y_{1,k} & Y_{2,k} \\ 0 & Y_{1,k}^T \end{bmatrix}, \quad Z_k = \begin{bmatrix} Z_{1,k} & Z_{2,k} \\ 0 & Z_{1,k}^T \end{bmatrix},$$

where $Y_{2,k}^T = -Y_{2,k}$, $Z_{2,k}^T = -Z_{2,k}$, and

$$\begin{aligned} Y_{1,k+1} &= -Y_{1,k} Z_{1,k}^{-1} Y_{1,k}, \\ Y_{2,k+1} &= -Y_{1,k} Z_{1,k}^{-1} Y_{2,k} + Y_{1,k} Z_{1,k}^{-1} Z_{2,k} Z_{1,k}^{-T} Y_{1,k}^T - Y_{2,k} Z_{1,k}^{-T} Y_{1,k}^T, \\ Z_{1,k+1} &= Z_{1,k} + 2Y_{1,k+1}, \\ Z_{2,k+1} &= Z_{2,k} + 2Y_{2,k+1}. \end{aligned}$$

4. Numerical experiments. We have performed several numerical experiments on an Athlon XP 2400, CPU at 2002 MHz, using MATLAB. We have compared our algorithm based on cyclic reduction (CR) with

1. the Denman–Beavers (DB) iteration [5]

$$\begin{aligned} Y_0 = A,\ Z_0 = I, \\ Y_{k+1} = (Y_k + Z_k^{-1})/2, \\ Z_{k+1} = (Z_k + Y_k^{-1})/2,\ k = 0, 1, 2, \ldots \end{aligned}$$

such that $Y_k \to A^{1/2}$, $Z_k \to A^{-1/2}$ as $k \to \infty$;

2. the scaled DB iteration [12]

$$\begin{aligned} Y_0 &= A,\ Z_0 = I,\\ \gamma_k &= |\det(Y_k)\det(Z_k)|^{-1/(2n)},\\ Y_{k+1} &= (\gamma_k Y_k + \gamma_k^{-1} Z_k^{-1})/2,\\ Z_{k+1} &= (\gamma_k Z_k + \gamma_k^{-1} Y_k^{-1})/2,\ k = 0, 1, 2, \ldots \end{aligned}$$

such that $Y_k \to A^{1/2}$, $Z_k \to A^{-1/2}$ as $k \to \infty$;

3. the method based on Padé approximations [15, 16, 12]

$$\begin{aligned} Y_0 &= A,\ Z_0 = I,\\ Y_{k+1} &= \tfrac{1}{p} Y_k \textstyle\sum_{i=1}^{p} \frac{1}{\xi_i} (Z_k Y_k + \alpha_i^2 I)^{-1},\\ Z_{k+1} &= \tfrac{1}{p} Z_k \textstyle\sum_{i=1}^{p} \frac{1}{\xi_i} (Y_k Z_k + \alpha_i^2 I)^{-1},\ k = 0, 1, 2, \ldots, \end{aligned}$$

where $p \geq 1$ is a chosen integer,

$$\xi_i = \frac{1}{2}\left(1 + \cos\frac{(2i-1)\pi}{2p}\right),\ \alpha_i^2 = \frac{1}{\xi_i} - 1,\quad i = 1, \ldots, p,$$

and $Y_k \to A^{1/2}$, $Z_k \to A^{-1/2}$ as $k \to \infty$;

4. the method based on scaled Padé approximations [12]

$$\begin{aligned} Y_0 &= A,\ Z_0 = I,\\ \gamma_k &= |\det(Y_k)\det(Z_k)|^{-1/(2n)},\\ Y_{k+1} &= \tfrac{1}{p} \gamma_k Y_k \textstyle\sum_{i=1}^{p} \frac{1}{\xi_i} (\gamma_k^2 Z_k Y_k + \alpha_i^2 I)^{-1},\\ Z_{k+1} &= \tfrac{1}{p} \gamma_k Z_k \textstyle\sum_{i=1}^{p} \frac{1}{\xi_i} (\gamma_k^2 Y_k Z_k + \alpha_i^2 I)^{-1},\ k = 0, 1, 2, \ldots; \end{aligned}$$

5. the method based on polar decomposition (PD) [8], when A is real symmetric positive definite:
   - compute $A = R^T R$, the Cholesky decomposition;
   - compute $U = X_\infty$ from $X_0 = R$, $X_{k+1} = (X_k + X_k^{-T})/2$, $k = 0, 1, \ldots$;
   - set $A^{1/2} = U^T R$;
6. the method based on polar decomposition with scaling (PD scaled) [8], when A is real symmetric positive definite:
   - compute $A = R^T R$, the Cholesky decomposition;
   - compute $U = X_\infty$ from $X_0 = R$, $X_{k+1} = (\gamma_k X_k + \gamma_k^{-1} X_k^{-T})/2$, $\gamma_k = \left(\frac{\|X_k^{-1}\|_1\ \|X_k^{-1}\|_\infty}{\|X_k\|_1 \|X_k\|_\infty}\right)^{1/4}$, $k = 0, 1, \ldots$;
   - set $A^{1/2} = U^T R$;
7. The `sqrtm` function of MATLAB, which uses the Schur algorithm [3, 10].

For each test matrix we have reported

1. the condition number estimate $\mu(A^{1/2})$ of the matrix square root provided by the MATLAB function `sqrtm`;
2. the relative residual

$$\frac{||X^2 - A||}{||A||};$$

TABLE 4.1
Prolate matrix, $n = 20$, $\mu(A^{1/2}) = 3.7 \cdot 10^6$.

Method	Iters.	Residual	Error
CR	25	1.7e-15	9.3e-10
Padé unscaled $p = 1$	17	4.6e-12	4.7e-7
$p = 2$	9	2.8e-12	3.3e-7
$p = 3$	7	2.6e-12	3.1e-7
$p = 4$	6	2.0e-12	3.3e-7
Padé scaled $p = 1$	12	3.7e-10	1.9e-10
$p = 2$	8	2.9e-10	1.5e-10
$p = 3$	6	1.4e-10	8.6e-11
$p = 4$	5	8.6e-11	2.0e-10
DB unscaled	5	9.7e-04	2.4e-2
DB scaled	11	1.2e-06	5.7e-7
PD unscaled	17	5.8e-11	3.0e-6
PD scaled	8	1.6e-14	7.4e-12
`sqrtm`	*	1.8e-15	1.3e-10

3. the number of iterations needed to reach that residual; we stopped the iterations when the residual stopped decreasing significantly;
4. the relative error

$$\frac{||X - A^{1/2}||}{||A^{1/2}||};$$

here $A^{1/2}$ has been computed by means of CR by using the Symbolic Computation Toolbox with a precision floating point arithmetic with 40 decimal digits accuracy.

Here $||X||$ is the Frobenius norm of the matrix X.

Test 1 (Prolate matrix). We used the `prolate` function of the MATLAB `gallery` test matrices, which generates the Prolate matrix. It is a symmetric, ill-conditioned Toeplitz matrix, which depends on an input parameter w. If $0 < w < 0.5$, then the Prolate matrix is positive definite, its eigenvalues are distinct, lie in $(0, 1)$, and tend to cluster around 0 and 1. We used the default value of w, i.e., $w = 0.25$ and the size $n = 20$; in this case the spectral condition number of A is $\kappa_2(A) = 5.7 \cdot 10^{13}$. The results are reported in Table 4.1. The smallest residuals are obtained by the CR algorithm, PD scaled, and the `sqrtm` function. Also the errors of PD scaled are good. Since the condition number of $A^{1/2}$ is of the order of 10^6, we cannot expect a residual better than 10^{-10}. Thus all the results, except the ones obtained with the DB iteration, are acceptable. The poor performance of the latter algorithm is due to the ill-conditioning of the matrix A, which must be inverted at the first step. The CR algorithm, in terms of accuracy, seems insensitive to the ill-conditioning of A. The large number of iterations is due to the fact that A is close to a singular matrix, and that CR converges linearly in the case where A is singular. For larger values of n the residual of CR remains unchanged; the residual of (scaled) DB and Padé methods grows with n. It is also interesting to point out that the residual of CR, as a function of the number of iterations, reaches a minimum and after that it does not change in the subsequent iterations; for DB and Padé methods, the residual can grow much in the subsequent iterations.

TABLE 4.2
Frobenius matrix, $p(x) = (x-2)(x-5)\left((x+2)^2+\epsilon\right)$, $\epsilon = 10^{-8}$, $\mu(A^{1/2}) = 1.5 \cdot 10^{10}$.

Method	Iters.	Residual	Error
CR	19	7.5e-03	2.6e-5
Padé unscaled $p=1$	24	7.7e-02	1.9e-5
$p=2$	10	2.2e-03	5.0e-5
$p=3$	12	6.1e-02	8.5e-6
$p=4$	6	5.9e-03	4.2e-5
Padé scaled $p=1$	15	3.2e-02	2.0e-5
$p=2$	15	7.9e-02	1.8e-5
$p=3$	12	2.9e-02	6.4e-7
$p=4$	19	5.4e-03	1.5e-5
DB unscaled	20	2.0e-05	1.9e-9
DB scaled	13	1.4e-05	2.0e-9
`sqrtm`	*	6.3e-9	2.1e-8

Test 2 (Frobenius matrix). For this test A is the 4×4 Frobenius matrix whose characteristic polynomial is $p(x) = \sum_{i=0}^{4} p_i x^i = (x-2)(x-5)\left((x+2)^2+\epsilon\right)$, i.e.,

$$A = \begin{bmatrix} -p_3 & -p_2 & -p_1 & -p_0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

where ϵ is a small fixed real positive number. The eigenvalues of the matrix A are the zeros of $p(x)$. In particular A has two complex conjugate eigenvalues $-2 \pm i\sqrt{\epsilon}$ close to the negative real axis. The matrix A is well conditioned ($\mu(A) \approx 10$), while $A^{1/2}$ is ill-conditioned when ϵ is small. In this case the residuals are large, due to the ill-conditioning of $A^{1/2}$, and the DB method performs better than the other algorithms. The results are reported in Table 4.2.

We tried also with the Frobenius matrix associated with the polynomial $p(x) = (x-2)(x-5)\left((x+1)^2+\epsilon\right)$. In this case A has two complex conjugate eigenvalues $-1 \pm i\sqrt{\epsilon}$ close to -1; thus the matrix $I+A$, which must be inverted at the first step of CR, is close to a singular matrix when ϵ is small. We tried with $\epsilon = 10^{-5}$. The results are reported in Table 4.3. In this case the DB and Padé methods perform better than the CR algorithm, which suffers from the ill-conditioning of the matrix $I+A$. This drawback could be overcome by computing the matrix square root of αA, where α is a positive real number such that $I + \alpha A$ is well conditioned, and then by scaling by $\alpha^{1/2}$ the obtained approximation.

Test 3 (Dorr matrix). This is the matrix generated by the `dorr` function of the MATLAB `gallery` test matrices. It is a row diagonally dominant, tridiagonal matrix that is ill-conditioned for small values of the input argument $\theta \geq 0$. We performed experiments with $\theta = 10^{-7}$ and size $n = 10$. The numerical results are reported in Table 4.4. We observe that the CR algorithm provides the lowest residual, and the approximation provided by the DB method is very poor.

Test 4 (Scaled matrix). For this test $A = \alpha R$, where α is a positive real number, and R is the randomly generated 5×5 matrix

$$R = \begin{bmatrix} 0.20277 & 0.015274 & 0.41865 & 0.83812 & 0.50281 \\ 0.19872 & 0.74679 & 0.84622 & 0.019640 & 0.70947 \\ 0.60379 & 0.44510 & 0.52515 & 0.68128 & 0.42889 \\ 0.27219 & 0.93181 & 0.20265 & 0.37948 & 0.30462 \\ 0.19881 & 0.46599 & 0.67214 & 0.83180 & 0.18965 \end{bmatrix}.$$

TABLE 4.3
Frobenius matrix, $p(x) = (x-2)(x-5)\left((x+1)^2+\epsilon\right)$, $\epsilon = 10^{-5}$, $\mu(A^{1/2}) = 1.2 \cdot 10^6$.

Method	Iters.	Residual	Error
CR	13	1.4e-03	3.2e-7
Padé unscaled $p=1$	14	2.6e-07	1.2e-7
$p=2$	12	2.9e-09	7.2e-12
$p=3$	6	5.8e-08	1.1e-8
$p=4$	8	3.6e-10	6.3e-12
Padé scaled $p=1$	12	7.3e-09	5.6e-12
$p=2$	8	6.4e-09	6.3e-11
$p=3$	5	9.2e-09	3.8e-11
$p=4$	4	3.3e-09	4.5e-11
DB unscaled	15	5.8e-12	3.1e-12
DB scaled	12	2.4e-12	1.4e-12
`sqrtm`	*	2.4e-12	1.4e-12

TABLE 4.4
Dorr matrix, $n = 10$, $\mu(A^{1/2}) = 3.0 \cdot 10^8$.

Method	Iters.	Residual	Error
CR	25	7.0e-16	2.6e-8
Padé unscaled $p=1$	6	1.7e-15	5.1e-9
$p=2$	3	1.9e-15	5.1e-9
$p=3$	2	2.5e-15	5.1e-9
$p=4$	2	1.7e-15	5.1e-9
Padé scaled $p=1$	11	6.4e-4	3.2e-4
$p=2$	20	6.4e-8	5.3e-8
$p=3$	9	6.3e-8	4.3e-8
$p=4$	7	9.6e-9	8.6e-9
DB unscaled	1	1.1	4.5e-1
DB scaled	11	8.7e-3	9.3e-3
`sqrtm`	*	2.8e-15	1.2e-8

We have compared the number of iterations needed to reach the minimum residual for small and large values of α. We have applied CR, Padé and scaled Padé (with $p = 2$), DB, and scaled DB. The number of iterations are reported in Figure 4.1. We observe that the scaling of DB and Padé methods keeps constant the number of iterations. For the nonscaled methods the number of iterations reaches a minimum for a certain $\bar{\alpha}$, and then it grows for larger and smaller values of $\bar{\alpha}$. The minimum number of iterations of CR coincides with the number of iterations of DB scaled.

Test 5 (Neumann matrix). We tested our algorithm on the 16×16 matrix generated by the `neumann` function of the MATLAB `gallery` test matrices. It is the matrix resulting from discretizing the Neumann problem with the usual five point operator on a regular mesh. The matrix has a one-dimensional null space with null vector the vector of 1s. The CR algorithm shows a linear convergence and reaches the residual 9.6e-16 in 24 iterations, and the error of the approximation is 1.6e-8; this confirms relation (3.11), which holds when A is singular. The matrix square root computed with the function `sqrtm` has a residual 2.9e-15 and an error 3.8e-9 and has complex entries, with imaginary part of magnitude 10^{-9}. The Padé method, with $p = 1$, reaches the residual 2.6e-15 after 6 iterations, and the error is 2.5e-10. A drawback of Padé algorithm which we have observed in this example is that the residual reaches a minimum value at the sixth iteration, and then it starts to grow; for instance, after 60 iterations the residual is 1.0e-7. This problem is not encountered for the CR method.

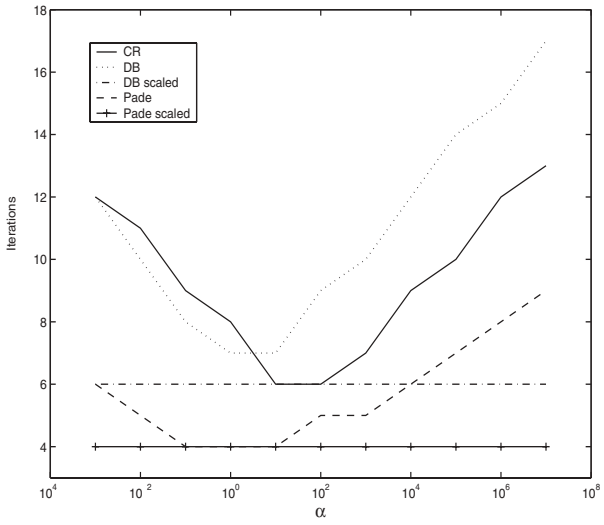


FIG. 4.1. *Scaled matrix: number of iterations*

TABLE 4.5
Matrix with prescribed singular values, $n = 10$, $\sigma_1 \approx \sqrt{2}$, $\sigma_{10} \approx 0$, $\mu(A^{1/2}) = 3.1 \cdot 10^4$.

Method	Iters.	Residual	Error
CR	15	1.4e-15	1.5e-12
Padé unscaled $p = 1$	13	2.0e-12	4.5e-10
$p = 2$	7	6.7e-13	3.4e-13
$p = 3$	6	8.3e-13	6.0e-13
$p = 4$	5	6.8e-13	6.5e-13
Padé scaled $p = 1$	10	1.3e-12	5.3e-13
$p = 2$	6	8.5e-13	9.3e-13
$p = 3$	5	8.6e-13	5.2e-13
$p = 4$	4	7.5e-13	3.8e-13
DB unscaled	13	8.4e-10	6.5e-10
DB scaled	10	2.2e-10	1.3e-10
`sqrtm`	*	3.9e-15	5.4e-13

Test 6 (matrix with prescribed singular values). By using the function `randcolu` of the MATLAB `gallery` test matrices, we have generated the 10×10 random matrix A having columns of unit 2-norm and having singular values $\sigma_1 = \sqrt{2 - \epsilon^2}$, $\sigma_2 = 1, \ldots, \sigma_9 = 1$, $\sigma_{10} = \epsilon$, where ϵ is a small real positive number. Thus σ_{10} is close to zero, while σ_1 is close to $\sqrt{2}$. We have chosen $\epsilon = 10^{-7}$. In Table 4.5 we have reported the results. The less accurate approximations are obtained by DB and the Padé algorithm with $p = 1$, the remaining algorithms provide small residuals and small errors.

We have also tested the same algorithms on the matrix A/ϵ, which has as singular values $\sigma_1 = \epsilon^{-1}\sqrt{2 - \epsilon^2}$, $\sigma_2 = \epsilon^{-1}, \ldots, \sigma_9 = \epsilon^{-1}$, $\sigma_{10} = 1$; thus σ_1 is large, while σ_{10} is equal to 1. As before, we have chosen $\epsilon = 10^{-7}$. For this matrix the effect of scaling of the DB and Padé methods is evident. In fact, the scaling considerably reduces the number of iterations.

By using the same function `randcolu` we have also generated the 10×10 matrix A having singular values $\sigma_1 = \sqrt{1 + \epsilon}$, $\sigma_2 = 1, \ldots, \sigma_9 = 1$, $\sigma_{10} = \sqrt{1 - \epsilon}$, where ϵ is a

small real number. Thus $\sigma_1 \approx \sigma_{10}^{-1}$. We have chosen $\epsilon = 10^{-5}$. This does not seem a critical case, since all the tested algorithms provide very small residuals and errors in few iterations.

5. Conclusions. We have given a new functional interpretation of the matrix square root of a matrix A, in terms of the inverse of a suitable matrix Laurent polynomial $R(z)$. This interpretation has allowed us to reduce the computation of the matrix square root to computing the constant block coefficient of $R(z)^{-1}$. For this purpose we have applied cyclic reduction, which, from the several numerical experiments, provides very accurate approximations of $A^{1/2}$. In particular, the cyclic reduction algorithm seems insensitive to the ill-conditioning of A, while other methods may fail to converge if A is ill-conditioned. In fact, cyclic reduction converges also in the important case where A is singular. An open question, which we are investigating, is to understand if a similar functional interpretation holds also for the matrix pth root, for $p > 2$. This would open the way to new algorithms for computing the matrix pth root.

Acknowledgments. The author wishes to thank N. J. Higham and the anonymous referees for providing useful suggestions to improve the presentation of this paper, and D. A. Bini and L. Gemignani for the useful discussions. In particular, the relation (3.9) with Newton's method was observed by L. Gemignani.

REFERENCES

[1] D. A. Bini, L. Gemignani, and B. Meini, *Computations with infinite Toeplitz matrices and polynomials*, Linear Algebra Appl., 343/344 (2002), pp. 21–61.
[2] D. A. Bini and B. Meini, *On the solution of a nonlinear matrix equation arising in queueing problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 906–926.
[3] Å. Björck and S. Hammarling, *A Schur method for the square root of a matrix*, Linear Algebra Appl., 52/53 (1983), pp. 127–140.
[4] B. L. Buzbee, G. H. Golub, and C. W. Nielson, *On direct methods for solving Poisson's equation*, SIAM J. Numer. Anal., 7 (1970), pp. 627–656.
[5] E. D. Denman and A. N. Beavers, Jr., *The matrix sign function and computations in systems*, Appl. Math. Comput., 2 (1976), pp. 63–94.
[6] C. R. DePrima and C. R. Johnson, *The range of $A^{-1}A^*$ in* GL(n, C), Linear Algebra Appl., 9 (1974), pp. 209–222.
[7] 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.
[8] N. J. Higham, *Computing the polar decomposition—with applications*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1160–1174.
[9] N. J. Higham, *Newton's method for the matrix square root*, Math. Comp., 46 (1986), pp. 537–549.
[10] N. J. Higham, *Computing real square roots of a real matrix*, Linear Algebra Appl., 88/89 (1987), pp. 405–430.
[11] N. J. Higham, *The matrix sign decomposition and its relation to the polar decomposition*, Linear Algebra Appl., 212/213 (1994), pp. 3–20.
[12] N. J. Higham, *Stable iterations for the matrix square root*, Numer. Algorithms, 15 (1997), pp. 227–242.
[13] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1994.
[14] B. Iannazzo, *A note on computing the matrix square root*, Calcolo, 40 (2003), pp. 273–283.
[15] C. Kenney and A. J. Laub, *Rational iterative methods for the matrix sign function*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 273–291.
[16] C. S. Kenney and A. J. Laub, *A hyperbolic tangent identity and the geometry of Padé sign function iterations*, Numer. Algorithms, 7 (1994), pp. 111–128.
[17] P. Laasonen, *On the iterative solution of the matrix equation $AX^2 - I = 0$*, Math. Tables Aids Comput., 12 (1958), pp. 109–116.
[18] L. Lu and C. E. M. Pearce, *On the square-root method for continuous-time algebraic Riccati equations*, J. Austral. Math. Soc. Ser. B, 40 (1999), pp. 459–468.

[19] C. F. Van Loan, *A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix*, Linear Algebra Appl., 61 (1984), pp. 233–251.

[20] H. G. Xu and L. Z. Lu, *Properties of a quadratic matrix equation and the solution of the continuous-time algebraic Riccati equation*, Linear Algebra Appl., 222 (1995), pp. 127–145.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 2, pp. 377–389

SPECTRAL PROPERTIES OF THE HERMITIAN AND SKEW-HERMITIAN SPLITTING PRECONDITIONER FOR SADDLE POINT PROBLEMS*

VALERIA SIMONCINI† AND MICHELE BENZI‡

Abstract. In this paper we derive bounds on the eigenvalues of the preconditioned matrix that arises in the solution of saddle point problems when the Hermitian and skew-Hermitian splitting preconditioner is employed. We also give sufficient conditions for the eigenvalues to be real. A few numerical experiments are used to illustrate the quality of the bounds.

Key words. saddle point problems, iterative methods, preconditioning, eigenvalues

AMS subject classifications. 65F10, 65N22, 65F50, 15A06

DOI. 10.1137/S0895479803434926

1. Introduction.

We are given the saddle point problem

$$\begin{pmatrix} A & B^T \\ -B & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f \\ -g \end{pmatrix}, \qquad \text{or} \qquad \mathcal{A}x = b \tag{1.1}$$

with $A \in \mathbb{R}^{n\times n}$ symmetric positive semidefinite and $B \in \mathbb{R}^{m\times n}$ with $\operatorname{rank}(B) = m \leq n$. We assume that the null spaces of A and B have trivial intersection, which implies that $\mathcal{A}$ is nonsingular. We set

$$\mathcal{H} = \begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix} \qquad \mathcal{S} = \begin{pmatrix} 0 & B^T \\ -B & 0 \end{pmatrix},$$

so that $\mathcal{A} = \mathcal{H}+\mathcal{S}$. We consider the preconditioner $\mathcal{P} = (2\alpha)^{-1}(\mathcal{H}+\alpha I)(\mathcal{S}+\alpha I)$, with real $\alpha > 0$, and we study the eigenvalue problem associated with the preconditioned matrix, that is,

$$(\mathcal{H}+\mathcal{S})x = \eta(2\alpha)^{-1}(\mathcal{H}+\alpha I)(\mathcal{S}+\alpha I)x. \tag{1.2}$$

This preconditioner has been studied in a somewhat more general setting in [4], motivated by the paper [1]. Letting $D(1,1) := \{z \in \mathbb{C}\,;\, |z-1| < 1\}$, it was shown in [4] that the spectrum of the preconditioned matrix satisfies $\sigma(\mathcal{P}^{-1}\mathcal{A}) \subset \overline{D(1,1)}\backslash\{0\}$. Furthermore, $\sigma(\mathcal{P}^{-1}\mathcal{A}) \subset D(1,1)$ if A is positive definite. Some rather special cases (including the case $A = I$) have been studied in [2, 3]. The purpose of this paper is to provide more refined inclusion regions for the spectrum of $\mathcal{P}^{-1}\mathcal{A}$ for saddle point problems of the form (1.1). Most of our bounds are in terms of the extreme eigenvalues and singular values of the blocks A and B, respectively. Although these quantities may be difficult to estimate, our results can be used to explain why small values of α usually give the best results in terms of convergence rates. For instance, we show

*Received by the editors September 17, 2003; accepted for publication (in revised form) by D. Szyld December 29, 2003; published electronically November 17, 2004.
http://www.siam.org/journals/simax/26-2/43492.html

†Dipartimento di Matematica, Università di Bologna, P.zza di Porta S. Donato, 5, I-40127 Bologna, Italy and IMATI-CNR, Pavia, Italy (valeria@dm.unibo.it).

‡Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322 (benzi@mathcs.emory.edu). The work of this author was supported in part by National Science Foundation grant DMS-0207599.

that sufficiently small values of α always result in preconditioned matrices having a real spectrum consisting of two tight clusters.

Throughout the paper, we write M^T for the transpose of a matrix M and u^* for the conjugate transpose of a complex vector u. Also, $A > 0$ ($A \geq 0$) means that matrix A is symmetric positive definite (respectively, semidefinite).

2. Spectral bounds. In this section we provide bounds for the eigenvalues of the preconditioned matrix.

In the following we shall use the fact that A is symmetric positive semidefinite, so that

$$0 \leq \lambda_n \leq \frac{u^*Au}{u^*u} \leq \lambda_1 \qquad \forall\ u \in \mathbb{C}^n,\ u \neq 0, \tag{2.1}$$

where λ_n, λ_1 are the smallest and largest eigenvalues of A, respectively. Moreover, we denote by $\sigma_1, \ldots, \sigma_m$ the decreasingly ordered singular values of B.

The spectrum of the preconditioned matrix can be more easily analyzed by means of a particular spectral mapping, which we introduce next. We shall then derive estimates for the location of the eigenvalues of (1.2).

We first observe that $(\mathcal{H} + \alpha I)(\mathcal{S} + \alpha I) = \mathcal{H}\mathcal{S} + \alpha(\mathcal{H} + \mathcal{S}) + \alpha^2 I$. By collecting the terms with $(\mathcal{H} + \mathcal{S})$ we can write the eigenvalue problem (1.2) as

$$\left(1 - \frac{1}{2}\eta\right)(\mathcal{H} + \mathcal{S})x = \frac{\eta\alpha}{2}\left(I + \frac{1}{\alpha^2}\mathcal{H}\mathcal{S}\right)x. \tag{2.2}$$

If $1 - \frac{1}{2}\eta = 0$, then $\eta = 2$. For $1 - \frac{1}{2}\eta \neq 0$ we set

$$\theta := \frac{\eta\alpha}{2 - \eta}, \qquad \text{from which} \qquad \eta = 2 - \frac{2\alpha}{\theta + \alpha} = \frac{2\theta}{\theta + \alpha}. \tag{2.3}$$

Therefore, (2.2) can be written as $(\mathcal{H} + \mathcal{S})x = \theta\left(I + \frac{1}{\alpha^2}\mathcal{H}\mathcal{S}\right)x$.

By explicitly writing the term $\mathcal{H}\mathcal{S}$, the eigenproblem above becomes

$$\begin{pmatrix} A & B^T \\ -B & 0 \end{pmatrix} x = \theta \begin{pmatrix} I & \frac{1}{\alpha^2}AB^T \\ 0 & I \end{pmatrix} x, \qquad \text{or} \qquad \mathcal{A}x = \theta\mathcal{G}x,$$

where

$$\mathcal{G} := \begin{pmatrix} I & \frac{1}{\alpha^2}AB^T \\ 0 & I \end{pmatrix}.$$

The equivalent eigenproblem $\mathcal{G}^{-1}\mathcal{A}x = \theta x$ can be explicitly written as

$$\begin{pmatrix} A + \frac{1}{\alpha^2}AB^TB & B^T \\ -B & 0 \end{pmatrix} x = \theta x. \tag{2.4}$$

Therefore, the two eigenproblems (1.2) and (2.4) have the same eigenvectors, while the eigenvalues are related by (2.3). Our spectral analysis aims at describing the behavior of the spectrum of $\mathcal{G}^{-1}\mathcal{A}$, from which considerations on the spectrum of (1.2) can be derived. In the following, $\Im(\theta)$ and $\Re(\theta)$ denote the imaginary and real part of θ, respectively.

LEMMA 2.1. *Assume A is symmetric and positive semidefinite. Let $K := I + \frac{1}{\alpha^2}B^TB$. For each eigenpair $(\eta, [u; v])$ of (1.2), η either is $\eta = 2$ or can be written as $\eta = 2 - \frac{2\alpha}{\alpha+\theta}$, where $\theta \neq 0$ satisfies the following:*

1. *If* $\Im(\theta) \neq 0$, *then*

$$\Re(\theta) = \frac{1}{2}\frac{u^*KAKu}{u^*Ku}, \qquad |\theta|^2 = \frac{u^*KB^TBu}{u^*Ku}. \tag{2.5}$$

2. *If* $\Im(\theta) = 0$, *then*

$$\min\left\{\lambda_n, \frac{\sigma_m^2}{\lambda_1\left(1+\frac{\sigma_m^2}{\alpha^2}\right)}\right\} \leq \theta \leq \rho$$

where $\rho := \lambda_1(1+\frac{\sigma_1^2}{\alpha^2})$.

Proof. The first statement of the lemma was already shown by means of the mapping in (2.3). We are thus left with proving the estimates for θ. First of all, note that $\theta \neq 0$ or else $\eta = 0$, which is not possible since $\mathcal{P}^{-1}\mathcal{A}$ is nonsingular.

Let $x = [u; v] \neq 0$ be the complex eigenvector associated with θ. We explicitly observe that $K = I + \frac{1}{\alpha^2}B^TB$ is symmetric positive definite and that KB^TB is symmetric. We shall make use of the following properties of K,

$$\lambda_{\max}(K) = 1 + \frac{\sigma_1^2}{\alpha^2}, \qquad \lambda_{\min}(K) \geq 1, \tag{2.6}$$

where the inequality becomes an equality whenever B is not square. In addition,

$$\lambda_n \leq \frac{u^*KAKu}{u^*K^2u} \leq \lambda_1, \tag{2.7}$$

and using $KB^TB = \alpha^2(K^2 - K)$,

$$0 \leq \frac{u^*KB^TBu}{u^*K^2u} = \alpha^2\frac{u^*K^2u - u^*Ku}{u^*K^2u} = \alpha^2\left(1 - \frac{u^*Ku}{u^*K^2u}\right) \leq \alpha^2 \quad \forall u \neq 0. \tag{2.8}$$

The two matrix equations in (2.4) are given by

$$\left(A + \frac{1}{\alpha^2}AB^TB\right)u + B^Tv = \theta u, \tag{2.9}$$

$$-Bu = \theta v. \tag{2.10}$$

It must be $u \neq 0$; otherwise (2.10) would imply $\theta = 0$ or $v = 0$, neither of which can be satisfied. For $u \neq 0$ and $v = 0$, from (2.9), θ must satisfy $AKu = \theta u$ and $Bu = 0$. Since K is symmetric and positive definite, we can write $K^{\frac{1}{2}}AK^{\frac{1}{2}}\hat{u} = \theta\hat{u}$, $\hat{u} = K^{\frac{1}{2}}u$, from which it follows that θ is real and satisfies

$$0 < \theta \leq \lambda_1\|K^{\frac{1}{2}}\|^2 = \lambda_1\lambda_{\max}\left(I + \frac{1}{\alpha^2}B^TB\right) = \lambda_1\left(1 + \frac{\sigma_1}{\alpha^2}\right) = \rho.$$

We now assume $u \neq 0 \neq v$. Using (2.10), we write $v = -\theta^{-1}Bu$, which, substituted into (2.9), yields $\theta A(I + \frac{1}{\alpha^2}B^TB)u - B^TBu = \theta^2u$. By multiplying this equation from the left by u^*K we obtain

$$\theta u^*KAKu - u^*KB^TBu = \theta^2u^*Ku. \tag{2.11}$$

Let $\theta = \theta_1 + \imath\theta_2$. For A symmetric, the quadratic equation (2.11) has real coefficients so that its roots are given by

$$\theta_\pm = \frac{1}{2}\frac{u^*KAKu}{u^*Ku} \pm \sqrt{\frac{1}{4}\left(\frac{u^*KAKu}{u^*Ku}\right)^2 - \frac{u^*KB^TBu}{u^*Ku}}. \tag{2.12}$$

Eigenvalues with nonzero imaginary part arise if the discriminant is negative.

Case $\theta_2 \neq 0$. It must be

$$(u^*KAKu)^2 - 4(u^*Ku)(u^*KB^TBu) < 0, \tag{2.13}$$

and from (2.12) we get $\theta_1 = \frac{1}{2}\frac{u^*KAKu}{u^*Ku}$. By substituting θ_1 in (2.12), we obtain $\theta_2^2 + \theta_1^2 = \frac{u^*KB^TBu}{u^*Ku}$.

Case $\theta_2 = 0$. In this case, from (2.12) it follows that $\theta = \theta_1 > 0$. For $Bu = 0$, from (2.10) it follows that $v = 0$ ($\theta \neq 0$), and the reasoning for $v = 0$ applies.

We now assume that $Bu \neq 0$. We have

$$-\theta_1^2 u^*Ku + \theta_1 u^*KAKu = u^*KB^TBu > 0,$$

where the last inequality follows from (2.8). Since $\theta_1 > 0$, the inequality $\theta_1 u^*KAKu - \theta_1^2 u^*Ku > 0$ implies $u^*KAKu - \theta_1 u^*Ku > 0$, hence $\theta_1 < \lambda_1\lambda_{\max}(K) = \rho$.

To prove the lower bound on θ, write the equation (2.9) as $(AK - \theta I)u = -B^Tv$. If θ is an eigenvalue of AK, then $\theta \geq \lambda_n\lambda_{\min}(K) \geq \lambda_n$. Otherwise, $(AK - \theta I)$ is invertible, so that $u = -(AK - \theta I)^{-1}B^Tv$, which, substituted into (2.10), yields

$$B(AK - \theta I)^{-1}B^Tv = \theta v \Leftrightarrow BK^{-1}(A - \theta K^{-1})^{-1}B^Tv = \theta v. \tag{2.14}$$

Let $B^T = [W_1, W_2]\begin{bmatrix}\Sigma\\ 0\end{bmatrix}Q^T$ be the singular value decomposition of B^T, and note that

$$\begin{aligned} K &= [W_1, W_2]\begin{pmatrix} I + \frac{1}{\alpha^2}\Sigma^2 & 0\\ 0 & I\end{pmatrix}[W_1^T, W_2^T]^T,\\ BK^{-1} &= Q\begin{pmatrix}\Sigma(I + \frac{1}{\alpha^2}\Sigma^2)^{-1} & 0\end{pmatrix}[W_1^T, W_2^T]^T = QD^{-1}\Sigma W_1^T, \end{aligned}$$

where $D = I + \frac{1}{\alpha^2}\Sigma^2$. Problem (2.14) can be thus written as $QD^{-1}\Sigma W_1^T(A - \theta K^{-1})^{-1}W_1\Sigma Q^Tv = \theta v$, or, equivalently,

$$\Sigma W_1^T(A - \theta K^{-1})^{-1}W_1\Sigma w = \theta Dw, \qquad w = Q^Tv,$$

from which

$$W_1^T(A - \theta K^{-1})^{-1}W_1\widehat{w} = \theta\Sigma^{-1}D\Sigma^{-1}\widehat{w}, \qquad \widehat{w} = \Sigma w. \tag{2.15}$$

We multiply both sides from the left by $\widehat{w}^*$ and we notice that the left-hand side is positive for any $\widehat{w} \neq 0$. If $\theta \geq \lambda_{\min}(AK) \geq \lambda_n$, then λ_n is the sought-after lower bound. Assume now that $\theta < \lambda_{\min}(AK)$. Then, the matrix $A - \theta K^{-1}$ is symmetric and positive definite. Therefore,

$$\begin{aligned} \widehat{w}^*W_1^T(A - \theta K^{-1})^{-1}W_1\widehat{w} &\geq \lambda_{\min}((A - \theta K^{-1})^{-1})\|W_1\widehat{w}\|^2\\ &= \lambda_{\min}((A - \theta K^{-1})^{-1})\|\widehat{w}\|^2, \end{aligned} \tag{2.16}$$

and we have

$$\lambda_{\min}((A-\theta K^{-1})^{-1}) = \frac{1}{\lambda_{\max}(A-\theta K^{-1})} \geq \frac{1}{\lambda_1 - \theta\lambda_{\min}(K^{-1})} = \frac{1}{\lambda_1 - \frac{\theta}{\lambda_{\max}(K)}} = \frac{1}{\lambda_1 - \frac{\theta}{\tau}},$$

where $\tau := \lambda_{\max}(K) = (1+\frac{\sigma_1^2}{\alpha^2})$. This, together with (2.16), provides a lower bound for the left-hand side of (2.15). Using

$$\theta\widehat{w}^*\Sigma^{-1}D\Sigma^{-1}\widehat{w} = \theta\widehat{w}^*\left(\Sigma^{-2}+\frac{1}{\alpha^2}I\right)\widehat{w} \leq \theta\left(\frac{1}{\sigma_m^2}+\frac{1}{\alpha^2}\right)\|\widehat{w}\|^2$$

and recalling that $\lambda_1\tau - \theta > 0$, from (2.15) we obtain

$$\frac{1}{\lambda_1 - \frac{\theta}{\tau}} \leq \theta\left(\frac{1}{\sigma_m^2}+\frac{1}{\alpha^2}\right), \qquad \text{i.e.,} \qquad \frac{\theta^2}{\tau} + \frac{\sigma_m^2\alpha^2}{\alpha^2+\sigma_m^2} \leq \lambda_1\theta.$$

Since $\theta^2 > 0$, we get $\frac{\sigma_m^2\alpha^2}{\alpha^2+\sigma_m^2} \leq \lambda_1\theta$, and the final bound follows. ∎

The quantities in part 1 of the lemma can also be bounded with techniques similar to those for the real case. However, in the next theorem, we derive sharper bounds for complex η than those one would obtain by using estimates for complex θ.

THEOREM 2.2. *Under the hypotheses and notation of Lemma* 2.1*, the eigenvalues of problem* (1.2) *are such that the following hold:*

1. *If* $\Im(\eta) \neq 0$*, then*

$$\frac{(\alpha+\frac{1}{2}\lambda_n)\lambda_n}{3\alpha^2} < \Re(\eta) < \min\left\{2, \frac{4\alpha}{\alpha+\lambda_n}\right\}, \tag{2.17}$$

$$\frac{\lambda_n^2}{3\alpha^2+\frac{1}{4}\lambda_n^2} < |\eta|^2 \leq \frac{4\alpha}{\alpha+\alpha(1+\frac{\sigma_1^2}{\alpha^2})^{-1}+\lambda_n}. \tag{2.18}$$

2. *If* $\Im(\eta) = 0$*, then* $\eta > 0$ *and*

$$\min\left\{\frac{2\lambda_n}{\alpha+\lambda_n}, \frac{2\frac{\sigma_m^2}{\varrho}}{\alpha+\frac{\sigma_m^2}{\varrho}}\right\} \leq \eta \leq \frac{2\rho}{\alpha+\rho} < 2, \tag{2.19}$$

where $\varrho := \lambda_1(1+\frac{\sigma_m^2}{\alpha^2})$ *and* $\rho := \lambda_1(1+\frac{\sigma_1^2}{\alpha^2})$.

Proof. We have that η is real if and only if θ is real. Assume $\Im(\eta) \neq 0$ and write $\theta = \theta_1 + \imath\theta_2$. Recall that $\tau = (1+\frac{\sigma_1^2}{\alpha^2})$.

Using the definition of θ in (2.3) we obtain

$$\Re(\eta) = 2\frac{\alpha\theta_1 + |\theta|^2}{\alpha^2 + 2\alpha\theta_1 + |\theta|^2},$$

that is, $(\alpha^2 + 2\alpha\theta_1 + |\theta|^2)\Re(\eta) = 2\alpha\theta_1 + 2|\theta|^2$. We substitute the quantities in (2.5) to get $(\alpha^2 u^*Ku + \alpha u^*KAKu + u^*KB^TBu)\Re(\eta) = \alpha u^*KAKu + 2u^*KB^TBu$. Note that $\alpha^2 u^*Ku + u^*KB^TBu = \alpha^2 u^*K^2u$. We divide by $u^*K^2u > 0$ to obtain

$$\left(\alpha^2 + \alpha\frac{u^*KAKu}{u^*K^2u}\right)\Re(\eta) = \alpha\frac{u^*KAKu}{u^*K^2u} + 2\frac{u^*KB^TBu}{u^*K^2u}.$$

We recall that for $\Im(\eta) \neq 0$ relation (2.13) holds, which implies by (2.6) and (2.8)

$$\frac{(u^*KAKu)^2}{(u^*K^2u)^2} < 4\frac{(u^*Ku)}{u^*K^2u}\frac{(u^*KB^TBu)}{u^*K^2u} \leq 4\alpha^2 \tag{2.20}$$

and

$$\frac{(u^*KB^TBu)}{u^*K^2u} > \frac{1}{4}\frac{(u^*KAKu)^2}{(u^*K^2u)^2}\frac{(u^*K^2u)}{u^*Ku} \geq \frac{1}{4}\lambda_n^2. \tag{2.21}$$

Therefore, by applying (2.7), (2.20), and (2.8), we obtain

$$(\alpha^2 + \alpha\lambda_n)\Re(\eta) < \alpha(2\alpha) + 2\alpha^2 \Leftrightarrow \Re(\eta) < \frac{4\alpha}{\alpha + \lambda_n}.$$

By once more applying (2.20), (2.7), and (2.21), we also get

$$(\alpha^2 + \alpha(2\alpha))\Re(\eta) > \alpha\lambda_n + \frac{1}{2}\lambda_n^2 \Leftrightarrow \Re(\eta) > \frac{(\alpha + \frac{1}{2}\lambda_n)\lambda_n}{3\alpha^2},$$

which provide the upper and lower bounds for $\Re(\eta)$.

To complete the proof of the first statement, we write $|\eta|^2$ using (2.3) to obtain

$$(\alpha^2 + 2\alpha\theta_1)|\eta|^2 = (4 - |\eta|^2)|\theta|^2.$$

Substituting (2.5) as before and dividing by u^*K^2u, it yields

$$\left(\alpha^2\frac{u^*Ku}{u^*K^2u} + \alpha\frac{u^*KAKu}{u^*K^2u}\right)|\eta|^2 = (4 - |\eta|^2)\frac{u^*KB^TBu}{u^*K^2u}.$$

Note that $4 - |\eta|^2 > 0$. As before, we bound $|\eta|^2$ from both sides, keeping in mind (2.6), (2.7), (2.8), (2.21), and (2.20), to get

$$\left(\frac{1}{\tau}\alpha^2 + \alpha\lambda_n\right)|\eta|^2 \leq 4\alpha^2 - |\eta|^2\alpha^2 \Leftrightarrow |\eta|^2 \leq \frac{4\alpha}{\alpha + \alpha(1 + \frac{\sigma_1^2}{\alpha^2})^{-1} + \lambda_n},$$

and

$$(\alpha^2 + \alpha(2\alpha))|\eta|^2 > \frac{1}{4}\lambda_n^2(4 - |\eta|^2) \Leftrightarrow |\eta|^2 > \frac{\lambda_n^2}{3\alpha^2 + \frac{1}{4}\lambda_n^2}.$$

This completes the proof of the first part.

Assume now that η is real. Then, from the corresponding bound for real θ in Lemma 2.1 and the fact that $\eta = \phi(\theta) = \frac{2\theta}{\alpha+\theta}$ is a strictly increasing function of its argument, we obtain the desired bounds on η. ∎

A few comments are in order. We start by noticing that, in general, real eigenvalues η may well cover the whole open interval $(0, 2)$, depending on the parameter α. Our numerical experiments show that these bounds are indeed sharp for several values of α (cf. section 4).

Although much less sharp in general, we also found the bounds for eigenvalues with nonzero imaginary part of interest. The lower estimate for $|\eta|$ indicates that nonreal eigenvalues are not close to the origin, especially for small α. In addition, they are located in a section of an annulus as in Figure 2.1. We will see in Theorem 3.1

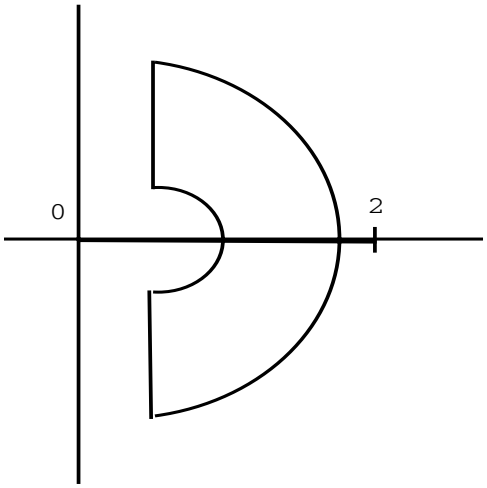


FIG. 2.1. *Inclusion region for the typical spectrum of the preconditioned matrix.*

that complex eigenvalues cannot arise for values of α smaller than one half the smallest eigenvalue of A.

Remark 2.1. We note that when A is positive definite, selecting $\alpha = \lambda_n$ provides constant bounds for the cluster of eigenvalues with nonzero imaginary part. Indeed, substituting $\alpha = \lambda_n$ in (2.17) and in (2.18) we obtain

$$\frac{1}{2} \leq \Re(\eta) < 2 \quad \text{and} \quad \frac{4}{13} \leq |\eta|^2 \leq \frac{4(\lambda_n^2 + \sigma_1^2)}{3\lambda_n^2 + 2\sigma_1^2} \leq \frac{4(\lambda_n^2 + \sigma_1^2)}{2\lambda_n^2 + 2\sigma_1^2} = 2.$$

For $\alpha \approx \lambda_n$ we expect to obtain similar bounds. This complex clustering seems to be relevant in the performance of the preconditioned iteration; cf. section 4.

3. Conditions for a real spectrum and clustering properties. We next show that under suitable conditions, the spectrum of the nonsymmetric preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ is real. We stress the fact that a real spectrum is a welcome property, because it enables the efficient use of short-recurrence Krylov subspace methods such as Bi-CGSTAB; see, e.g., [11, p. 139].

THEOREM 3.1. *Assume the hypotheses and notation of Lemma* 2.1 *hold and assume in addition that A is symmetric positive definite. If $\alpha \leq \frac{1}{2}\lambda_n$, then all eigenvalues η are real.*

Proof. We prove our assertion for the eigenvalues θ, from which the statement for η will follow. Let $x = [u; v]$ be an eigenvector associated with θ. For $u \neq 0, v = 0$ we already showed that the spectrum is real, while $u = 0$ implies $v = 0$, a contradiction. We now assume $u \neq 0 \neq v$.

The eigenvalues θ of (2.4) are the roots of equation (2.11), which can be expressed as in (2.12). These are all real if the discriminant is nonnegative. Equivalently,

$$\theta \in \mathbb{R} \quad \text{if} \quad (u^*KAKu)^2 \geq 4(u^*Ku)\,(u^*KB^TBu) \quad \forall u \neq 0.$$

Since $u^*K^2u > 0$ for $u \neq 0$, we write the problem above as

$$\theta \in \mathbb{R} \quad \text{if} \quad \frac{(u^*KAKu)^2}{(u^*K^2u)^2} \geq 4\frac{u^*Ku}{u^*K^2u}\,\frac{u^*KB^TBu}{u^*K^2u} \quad \forall u \neq 0.$$

We have $\frac{(u^*KAKu)^2}{(u^*K^2u)^2} \geq \lambda_n^2$, and $\frac{u^*Ku}{u^*K^2u} \leq \lambda_{\min}(K)^{-1} \leq 1$; see (2.6). Therefore, using (2.8), if $\alpha \leq \frac{1}{2}\lambda_n$, we have

$$\frac{(u^*KAKu)^2}{(u^*K^2u)^2} \geq \lambda_n^2 \geq 4 \cdot 1 \cdot \alpha^2 \geq 4 \frac{u^*Ku}{u^*K^2u} \frac{u^*KB^TBu}{u^*K^2u} \quad \forall u \neq 0. \tag{3.1}$$

The discriminant is nonnegative, therefore all roots of (2.12) are real, and so are the eigenvalues θ. ∎

The smallest eigenvalue of A can be increased by suitable scalings, thus enlarging the interval of α values leading to a real spectrum. Note, however, that multiplying (1.1) by a positive constant ω is equivalent to applying the Hermitian/skew-Hermitian splitting preconditioner with parameter $\hat{\alpha} := \sqrt{\omega}\alpha$ to the original, unscaled system.

Under additional assumptions on the spectrum of the block matrices, it is possible to provide a less strict condition on α. This is stated in the following corollary.

COROLLARY 3.2. *Under the hypotheses and notation of Theorem* 3.1*, assume that* $4\sigma_1^2 - \lambda_n^2 > 0$. *If* $\alpha \leq \frac{\lambda_n \sigma_1}{\sqrt{4\sigma_1^2 - \lambda_n^2}}$ *then all eigenvalues* η *are real.*

Proof. Using (2.8), we can write

$$\frac{u^*KB^TBu}{u^*K^2u} = \alpha^2 \left(1 - \frac{u^*Ku}{u^*K^2u}\right) \leq \alpha^2 \left(1 - \frac{1}{1 + \frac{\sigma_1^2}{\alpha^2}}\right) = \alpha^2 \frac{\sigma_1^2}{\alpha^2 + \sigma_1^2}.$$

Therefore, if $\lambda_n^2 \geq 4\alpha^2 \frac{\sigma_1^2}{\alpha^2 + \sigma_1^2}$, the bound equivalent to (3.1) follows. Moreover, we note that under the assumption that $4\sigma_1^2 - \lambda_n^2 > 0$,

$$\lambda_n^2 \geq 4\alpha^2 \frac{\sigma_1^2}{\alpha^2 + \sigma_1^2} \Leftrightarrow \alpha^2 \leq \frac{\lambda_n^2 \sigma_1^2}{4\sigma_1^2 - \lambda_n^2}. \qquad ∎$$

It is interesting to observe that if $\sigma_1^2 = \lambda_1$, the condition $4\sigma_1^2 - \lambda_n^2 > 0$ corresponds to the inequality

$$\frac{\lambda_1}{\lambda_n} > \frac{1}{4}\lambda_n,$$

which is easily satisfied since usually λ_n is small and λ_1 is much bigger than λ_n. Note that such a setting is very common in the Stokes problem, where A is a discretization of a (vector) Laplacian and BB^T can also be regarded as a discrete Laplacian.

The following result shows that the eigenvalues form two tight clusters as $\alpha \to 0$. This is an important property from the point of view of convergence of preconditioned Krylov subspace methods. This result extends and sharpens the clustering result obtained in [3] (using different tools) for the special case of Poisson's equation in saddle point form.

PROPOSITION 3.3. *Assume* A *is symmetric and positive definite. For sufficiently small* $\alpha > 0$*, the eigenvalues of* $\mathcal{P}^{-1}\mathcal{A}$ *cluster near zero and two. More precisely, for small* $\alpha > 0$*,* $\eta \in (0, \varepsilon_1) \cup (2 - \varepsilon_2, 2)$*, with* $\varepsilon_1, \varepsilon_2 > 0$ *and* $\varepsilon_1, \varepsilon_2 \to 0$ *for* $\alpha \to 0$.

Proof. We assume α is small, and in particular $\alpha \leq \frac{1}{2}\lambda_n$; therefore all eigenvalues are real. Let $[u; v]$ be an eigenvector of (2.4) and let $\theta_\pm$ be the roots of equation (2.11). These are given by (2.12). Collecting u^*Ku and dividing and multiplying (2.12) by $u^*K^2u > 0$, we obtain

$$\theta_\pm = \frac{u^*K^2u}{u^*Ku} \left(\frac{1}{2} \frac{u^*KAKu}{u^*K^2u} \pm \sqrt{\frac{1}{4} \left(\frac{u^*KAKu}{u^*K^2u} \right)^2 - \frac{u^*Ku}{u^*K^2u} \frac{u^*KB^TBu}{u^*K^2u}} \right) \equiv \frac{u^*K^2u}{u^*Ku} \nu_\pm.$$

We recall the bounds in (2.7) and (2.8), while $1 \le \frac{u^*K^2u}{u^*Ku} \le (1+\frac{\sigma_1^2}{\alpha^2})$ for any $u \neq 0$, with $(1+\frac{\sigma_1^2}{\alpha^2}) = O(\alpha^{-2})$ as $\alpha \to 0$. Moreover, $0 \le \frac{u^*Ku}{u^*K^2u}\frac{u^*KB^TBu}{u^*K^2u} \le \alpha^2$, so that $\frac{u^*Ku}{u^*K^2u}\frac{u^*KB^TBu}{u^*K^2u} \to 0$ as $\alpha \to 0$. We thus have $\nu_+ \to \frac{u^*KAKu}{u^*K^2u}$ as $\alpha \to 0$. Since $\frac{u^*KAKu}{u^*K^2u}$ is bounded independently of α, we also obtain

$$\nu_- = O\left(\frac{u^*Ku}{u^*K^2u}\frac{u^*KB^TBu}{u^*K^2u}\right) \qquad \text{for} \quad \alpha \to 0.$$

Therefore, $\theta_+ = O(\frac{u^*K^2u}{u^*Ku}) = O(\alpha^{-2})$ as $\alpha \to 0$, whereas $\theta_- = O(\frac{u^*KB^TBu}{u^*K^2u}) = O(\alpha^2)$ as $\alpha \to 0$. It thus follows that

$$\eta_+ = 2 - \frac{2}{1+\frac{\theta_+}{\alpha}} \to 2 \qquad \text{and} \qquad \eta_- = 2 - \frac{2}{1+\frac{\theta_-}{\alpha}} \to 0 \qquad \text{for} \quad \alpha \to 0. \qquad \square$$

We mention that the dependency of the "optimal" value of α on the mesh size h has been discussed, using Fourier analysis, in [3] for the case of Poisson's equation in first order system form, and in [5] for the case of the Stokes problem. In the first case one can choose α so as to have h-independent convergence, whereas in the second case there is a moderate growth in the number of iterations as $h \to 0$.

It is important to remark that the occurrence of a gap in the spectrum for small α can be deduced from known results for overdamped systems. Indeed, equation (2.11) stems from the quadratic eigenvalue problem

$$\theta^2 Ku - \theta KAKu + KB^TBu = 0.$$

The eigenproblem above has $2n$ eigenvalues, $n-m$ of which are zero, corresponding to the dimension of the null space of KB^TB. The remaining $n+m$ eigenvalues coincide with the eigenvalues of our problem (2.4). By introducing $\widetilde{\theta} = -\theta$, we obtain the quadratic symmetric eigenproblem (see [6])

$$\widetilde{\theta}^2 Ku + \widetilde{\theta} KAKu + KB^TBu = 0, \qquad K > 0, \quad KAK > 0, \quad KB^TB \ge 0.$$

It can be shown (see, e.g., [6, Theorem 13.1]) that if the discriminant is positive—that is, if $(u^*KAKu)^2 - 4(u^*Ku)(u^*KB^TBu) > 0$ for any $u \neq 0$—then all eigenvalues $\widetilde{\theta}$ are real and nonpositive. Moreover, the spectrum is split in two parts, each of which contains n eigenvalues.[1]

In our context, and in light of Proposition 3.3, the result above implies that m eigenvalues η will cluster towards zero, while n eigenvalues η will cluster around 2, for sufficiently small α.

4. Numerical experiments. In this section we present the results of a few numerical tests aimed at assessing the tightness of our bounds. The first problem we consider is a saddle point system arising from a finite element discretization of a model Stokes problem (leaky-lid driven cavity). This problem was generated using the IFISS software written by Howard Elman, Alison Ramage, and David Silvester [9]. Here $n = 578$, $m = 254$, $\lambda_n = 0.0763666$, $\lambda_1 = 3.949253$, $\sigma_1 = 0.247606661$, and $\sigma_m = 0.005319517$. Note that the B matrices (discrete divergence operators) generated by this software are rank deficient; we obtained a full rank matrix by dropping the two first rows of B.

[1] Note that in the statement of Theorem 13.1 in [6], matrix KB^TB is required to be positive definite rather than just semidefinite. However, the result is still true under the weaker assumption $KB^TB \ge 0$; see also the treatment in [10] and references therein.

TABLE 4.1
Real bounds in (2.19) *vs. actual eigenvalues, Stokes problem.*

α	Lower bound	$\eta_{\min}$	$\eta_{\max}$	Upper bound
0.001	0.00048902	0.00050629	1.9999	1.9999
0.01	0.00111635	0.00169724	1.9999	1.9999
0.1	0.00014289	0.00022355	1.9929	1.9929
0.2	0.00007160	0.00011205	1.9608	1.9608
0.3	0.00004775	0.00007473	1.9134	1.9135
0.4	0.00003582	0.00005606	1.8633	1.8635
0.5	0.00002866	0.00004485	1.8150	1.8154
0.6	0.00002388	0.00003738	1.7696	1.7702
0.7	0.00002047	0.00003204	1.7271	1.7278
0.8	0.00001791	0.00002803	1.6871	1.6880
0.9	0.00001592	0.00002492	1.6494	1.6504
1.0	0.00001433	0.00002243	1.6137	1.6147
2.0	0.00000717	0.00001121	1.3327	1.3344
5.0	0.00000287	0.00000449	0.8826	0.8838

TABLE 4.2
Bounds in (2.19) *vs. actual real eigenvalues, groundwater flow problem.*

α	Lower bound	$\eta_{\min}$	$\eta_{\max}$	Upper bound
0.001	0.181813	0.181818	2.000000	2.000000
0.01	0.285713	0.310869	1.999893	1.999971
0.05	0.064515	0.070481	1.985944	1.996341
0.1	0.032786	0.035865	0.137154	1.971127
0.3	0.011049	0.012099	0.047856	1.437903
0.5	0.006644	0.007277	0.028988	0.722331
1.0	0.003327	0.003645	0.014599	0.145003
3.0	0.001110	0.001217	0.004890	0.011648
5.0	0.000666	0.000730	0.002937	0.005078

In Table 4.1 we compare the lower and upper bounds given in Theorem 2.2 with the actual values of the smallest and largest eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$, which in this case are all real. One can see that the upper bound is always very tight and that the lower bound is fairly tight, especially for small values of α. For $\alpha \approx 0.01$ or smaller, the eigenvalues form two tight clusters near 0 and 2, containing m and n eigenvalues, respectively, as predicted by Proposition 3.3.

Next, we consider a saddle point system arising from the discretization of a groundwater flow problem using mixed-hybrid finite elements [7]. In the example at hand, $n = 270$, $m = 207$, $n + m = 477$, and $\mathcal{A}$ contains $1,746$ nonzeros. Here we have $\lambda_n = 0.0017$, $\lambda_1 = 0.010$, $\sigma_1 = 2.611$, and $\sigma_m = 0.19743$.

In this case there are nonreal eigenvalues (except for very small α). In Table 4.2 we compare the lower and upper bounds given in Theorem 2.2 with the actual values of the smallest and largest *real* eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ while in Tables 4.3 and 4.4 we provide the analogous results for the real part and modulus of the nonreal eigenvalues.

One can see that the location of the real eigenvalues is well detected with our bounds. In particular, the lower bound is very sharp, whereas the upper bound gets looser when the whole spectrum becomes complex ($\alpha \geq 0.05$), providing again good estimates for large values of α. The lower bounds suggest that the leftmost cluster will not be too close to zero, particularly for α between 10^{-3} and 10^{-2}, and it turns out that these values of α yield the best results (see below).

TABLE 4.3
Bounds in (2.17) *vs. actual real part of nonreal eigenvalues, groundwater flow problem.*

α	Lower bound	$\min \Re(\eta)$	$\max \Re(\eta)$	Upper bound
0.001	–	–	–	–
0.01	–	–	–	–
0.05	0.011296	1.823080	1.962387	2.000000
0.1	0.005602	1.571808	1.975776	2.000000
0.3	0.001857	0.608980	1.966375	2.000000
0.5	0.001113	0.274840	1.924906	2.000000
1.0	0.000556	0.078255	1.742401	2.000000
3.0	0.000185	0.009779	0.862083	2.000000
5.0	0.000111	0.003810	0.428775	2.000000

TABLE 4.4
Bounds in (2.18) *vs. actual modulus of nonreal eigenvalues, groundwater flow problem.*

α	Lower bound	$\min \lvert\eta\rvert$	$\max \lvert\eta\rvert$	Upper bound
0.001	–	–	–	–
0.01	–	–	–	–
0.05	0.019244	1.860113	1.963349	1.967129
0.1	0.009622	1.753875	1.977199	1.982111
0.3	0.003207	1.093125	1.979200	1.981669
0.5	0.001924	0.731979	1.959713	1.962379
1.0	0.000962	0.386709	1.865509	1.881779
3.0	0.000321	0.131260	1.312480	1.596393
5.0	0.000192	0.078883	0.925533	1.496510

Concerning nonreal eigenvalues, we observe that our bounds are generally not very sharp. The real part of the eigenvalues changes considerably as α varies, clustering on different regions of the interval $(0, 2)$. Our lower bounds on $\Re(\eta)$ are rather loose, although they get better for larger values of α; conversely, the upper bounds are tight for small α and loose for large α.

We conclude this section with the results of a few experiments that illustrate the convergence behavior of (full) GMRES [8] with Hermitian/skew-Hermitian splitting preconditioning; we refer to [4] for more extensive experimental results. The purpose of these experiments is to investigate the influence of the eigenvalue distribution, and in particular of the clustering that occurs as $\alpha \to 0$, on the convergence of GMRES. We also monitor the conditioning of the eigenvectors of the preconditioned matrix for different values of α.

In Table 4.5 we report a sample of results for both the Stokes and the groundwater flow problem, for different values of α (from tiny to fairly large). Here $\kappa_2(V) := \frac{\sigma_{\max}(V)}{\sigma_{\min}(V)}$ denotes the spectral condition number of the matrix of (normalized) eigenvectors of $\mathcal{P}^{-1}\mathcal{A}$, and "Its" denotes the corresponding number of preconditioned GMRES iterations (matrix-vector products) needed to reduce the initial residual by at least six orders of magnitude. For the Stokes problem, the condition number of the eigenvector matrix of the unpreconditioned $\mathcal{A}$ is $\kappa_2(V) = 6.94$. Without preconditioning, full GMRES converges in 199 iterations. For the (unpreconditioned) groundwater flow problem, it is $\kappa_2(V) = 1.37$ and GMRES stagnates.

Note that for both problems, the best results (in terms of GMRES iterations) are obtained for $\alpha = 0.005$, with generally good convergence behavior for α between 10^{-6} and 10^{-2}. Good performance is observed in particular for $\alpha \approx \lambda_n$, for which nonreal eigenvalues, when they occur, lie in a small region in the disc $D(1, 1)$ (cf. Remark 2.1).

TABLE 4.5
Conditioning of the eigenvectors and iteration count.

	Stokes		Groundwater flow	
α	$\kappa_2(V)$	Its	$\kappa_2(V)$	Its
10^{-12}	1.28E+18	> 200	4.31E+09	25
10^{-9}	1.31E+10	45	1.01E+08	17
10^{-6}	4.51E+08	41	1.41E+17	17
10^{-5}	3.30E+04	40	5.69E+00	17
10^{-4}	9.65E+03	40	1.23E+01	17
10^{-3}	1.48E+03	40	8.01E+00	13
0.005	1.16E+04	38	1.31E+03	11
0.01	1.18E+03	38	1.57E+04	13
0.03	7.63E+02	40	1.32E+01	17
0.05	2.68E+02	44	6.79E+01	19
0.07	2.26E+02	48	1.91E+01	20
0.1	6.05E+01	54	1.37E+01	26
0.3	3.55E+01	76	2.76E+00	67
0.5	4.38E+01	88	1.92E+00	109
0.7	2.88E+01	97	8.87E+00	> 200
1.0	1.77E+01	108	1.56E+00	> 200
5.0	3.33E+01	157	1.20E+00	> 200
10.0	6.44E+00	174	1.90E+00	> 200

The convergence rate remains fairly stable even for smaller values of α, but eventually it starts deteriorating as α approaches zero. It is likely that this is due to the fact that the preconditioner (and with it, the preconditioned matrix) becomes singular as $\alpha \to 0$. On the other hand, as $\alpha \to \infty$ the preconditioned matrix tends to the unpreconditioned one and the preconditioner becomes ineffective. Note that somewhat better results can be obtained by a suitable diagonal scaling of $\mathcal{A}$ (see [4]); however, no scaling was used here.

For both problems, $\kappa_2(V)$ appears to be very sensitive to changes in α, at least when α is small. This is in stark contrast with the rather smooth variation in the number of GMRES iterations. Overall, the condition number of the eigenvector matrix does not seem to have much influence on the convergence of GMRES.

5. Conclusions. In this paper we have provided bounds and clustering results for the spectra of preconditioned matrices arising from the application of the Hermitian/skew-Hermitian splitting preconditioner to saddle point problems. Numerical experiments have been used to illustrate the capability of our estimates to locate the actual spectral region. We have also shown that for small α, all the eigenvalues are real and fall in two clusters, one near 0 and the other near 2. Our bounds are especially sharp precisely for these values of α, which are those of practical interest. Indeed, our analysis suggests that the "best" value of α should be small enough so that the spectrum is clustered, but not so small that the preconditioned matrix is close to being singular. Numerical experiments confirm this, and it appears that when A is positive definite, $\alpha \approx \lambda_n(A)$ is generally a good choice.

Finally, we found a connection with the quadratic eigenvalue problems arising in the theory of overdamped systems; it is possible that exploitation of this connection may lead to further insight into the spectral properties of preconditioned saddle point problems.

Acknowledgment. We would like to thank Martin Gander for useful comments on an earlier draft of the paper.

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] Z. Z. Bai, G. H. Golub, and J. Y. Pan, *Preconditioned Hermitian and Skew-Hermitian Splitting Methods for Non-Hermitian Positive Semidefinite Linear Systems*, Technical Report SCCM-02-12, Scientific Computing and Computational Mathematics Program, Department of Computer Science, Stanford University, Stanford, CA, 2002.

[3] M. Benzi, M. J. Gander, and G. H. Golub, *Optimization of the Hermitian and skew-Hermitian splitting iteration for saddle-point problems*, BIT, 43 (2003), pp. 881–900.

[4] M. Benzi and G. H. Golub, *A preconditioner for generalized saddle point problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 20–41.

[5] M. Gander, *Optimization of a Preconditioner for Its Performance with a Krylov Method*, talk delivered at the Dagstuhl Seminar 03421 on Theoretical and Computational Properties of Matrix Algorithms, Dagstuhl, Germany, 2003 (http://www.dagstuhl.de/03421/).

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

[7] J. Maryška, M. Rozložník, and M. Tůma, *Mixed-hybrid finite element approximation of the potential fluid flow problem*, J. Comput. Appl. Math., 63 (1995), pp. 383–392.

[8] 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.

[9] D. Silvester, *Private communication*, 2002.

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

[11] H. A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, Cambridge Monogr. Appl. Comput. Math. 13, Cambridge University Press, Cambridge, UK, 2003.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 2, pp. 390–399

HERMITIAN MATRICES, EIGENVALUE MULTIPLICITIES, AND EIGENVECTOR COMPONENTS*

CHARLES R. JOHNSON† AND BRIAN D. SUTTON‡

Abstract. Given an n-by-n Hermitian matrix A and a real number λ, index i is said to be Parter (resp., neutral, downer) if the multiplicity of λ as an eigenvalue of the principal submatrix $A(i)$ is one more (resp., the same, one less) than that in A. In case the multiplicity of λ in A is at least 2 and the graph of A is a tree, there are always Parter vertices. Our purpose here is to advance the classification of vertices and, in particular, to relate classification to the combinatorial structure of eigenspaces. Some general results are given and then used to deduce some rather specific facts not otherwise easily observed. Examples are given.

Key words. eigenvalues, Hermitian matrix, multiplicity, Parter vertices

AMS subject classifications. 15A18, 05C50

DOI. 10.1137/S0895479802413649

1. Introduction.

Throughout this article, A will be an n-by-n Hermitian matrix and $A(i)$ its $(n-1)$-by-$(n-1)$ principal submatrix, resulting from deletion of row and column i, $i = 1, \ldots, n$. If $\lambda \in \mathbb{R}$ is an identified eigenvalue, we denote by $m_A(\lambda)$ its multiplicity as an eigenvalue of A. Because of the interlacing inequalities [HJ, Chap. 4], $|m_A(\lambda) - m_{A(i)}(\lambda)| \leq 1$, and all 3 values of $m_A(\lambda) - m_{A(i)}(\lambda)$ are possible. Because of recent work [JLD99, JLD02, JLD+, JLDS] and for historical reasons [P], we call the index i a *Parter* (resp., *downer*, *neutral*) index if $m_A(\lambda) - m_{A(i)}(\lambda) = -1$ (resp., 1, 0). In the event that the graph of A becomes relevant, recall that $G(A)$ is the graph on n vertices in which there is an edge between i and j if and only if the i, j entry of A is nonzero. By $\mathcal{H}(G)$ we denote the set of all Hermitian matrices whose graph is the given graph G; note that the diagonal entries are immaterial for belonging to the set $\mathcal{H}(G)$ (except that they are real). In discussing issues herein, we naturally identify vertices of $G(A)$ with indices, and induced subgraphs of $G(A)$ with principal submatrices of A, etc., in a benign way. It was shown in [P] and subsequent refinements [W, JLDS] that *for trees* there are always Parter vertices when $m_A(\lambda) \geq 2$ and further information about their existence when $m_A(\lambda) < 2$. As the location of Parter vertices in G is an important issue, our purpose here is to relate the classification of vertices (w.r.t. Parter, downer, and neutral) to the combinatorial structure of eigenspaces. However, the relationship may be of interest in both directions. As it turns out some of our observations do not depend upon the particular structure of $G(A)$.

If $m_A(\lambda) \geq 1$, denote the corresponding eigenspace by $E_A(\lambda)$. If $m_A(\lambda) = 0$, then we may, for convenience, adopt the convention that $E_A(\lambda)$ contains only the zero vector. In the event that entry i of x is 0 for every $x \in E_A(\lambda)$, we say that i is a *null vertex* (for A and λ); otherwise i is a *nonzero vertex*. Of course, there is an $x \in E_A(\lambda)$ whose support consists of all nonzero vertices.

*Received by the editors August 27, 2002; accepted for publication (in revised form) by I. S. Dhillon November 17, 2003; published electronically November 17, 2004.
http://www.siam.org/journals/simax/26-2/41364.html

†Department of Mathematics, College of William and Mary, Williamsburg, VA 23187-8795 (crjohnso@math.wm.edu).

‡Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139-4307 (bsutton@math.mit.edu). The research of this author was supported by a National Science Foundation Graduate Research Fellowship.

For trees, a useful characterization of when a vertex is Parter was demonstrated in [JLDS]. Removal of a vertex v of degree d in a tree leaves d induced subgraphs, each of which is a tree; such subgraphs are called branches at v and may be identified by the neighbors $u_1, \ldots, u_d$ of v. Vertex v is then Parter in the tree T if and only if there is an i, $1 \leq i \leq d$, such that u_i is a downer vertex in its branch (all w.r.t. some $\lambda \in \sigma(A), A \in \mathcal{H}(T)$).

We also use the notation $A(\{i_1, \ldots, i_k\})$ to indicate the $(n-k)$-by-$(n-k)$ principal submatrix of A resulting from deletion of rows and columns $i_1, \ldots, i_k$ from $A \in M_n$. In addition, $A[\{i_1, \ldots, i_k\}]$ denotes the k-by-k principal submatrix of A lying in the rows and columns indexed by $i_1, \ldots, i_k$. (When $k = 1$, we write $A[i]$.) When indices/vertices are deleted, we refer to the remaining vertices via their original numbers.

2. General result. From a simple and standard calculation, it is clear that when i is a null vertex, the structure of $E_A(\lambda)$ imparts a good deal of information about $E_{A(i)}(\lambda)$. Suppose, w.l.o.g., that $n = i$:

$$\left[\begin{array}{c|c} A(n) & a_{1n} \\ \hline a_{1n}^* & a_{nn} \end{array}\right] \left[\begin{array}{c} x \\ \hline 0 \end{array}\right] = \lambda \left[\begin{array}{c} x \\ \hline 0 \end{array}\right]. \tag{2.1}$$

(a_{1n} is a column vector, and a_{nn} is a scalar.) Then, $A(n)x = \lambda x$. This implies, in particular, that a null vertex is, at least, neutral. The converse is also valid.

THEOREM 2.1. *Let A be an n-by-n Hermitian matrix. Then, index i is null for A if and only if index i is either Parter or neutral.*

Our proof uses the following lemma. When taking principal submatrices, it is convenient to think of $E_{A(i)}(\lambda)$ as a subspace of $\mathbb{C}^n$. We define $E'_{A(i)}(\lambda)$ to be the $m_{A(i)}(\lambda)$-dimensional subspace of $\mathbb{C}^n$ formed by extending every vector of $E_{A(i)}(\lambda)$ by a zero in the ith coordinate.

LEMMA 2.2. *For an n-by-n Hermitian matrix A and an identified $\lambda \in \mathbb{R}$, we have the following:*

1. *If i is downer, then $E_A(\lambda) \supsetneq E'_{A(i)}(\lambda)$.*
2. *If i is neutral, then $E_A(\lambda) = E'_{A(i)}(\lambda)$.*
3. *If i is Parter, then $E_A(\lambda) \subsetneq E'_{A(i)}(\lambda)$.*

Proof. Assume w.l.o.g. that $i = n$ and $\lambda = 0$, and use the block decomposition of A shown in (2.1).

If a_{1n}^* is a linear combination of the rows of $A(n)$, then $E_A(0) \supseteq E'_{A(n)}(0)$. If a_{1n}^* is *not* a linear combination of the rows of $A(n)$, then sequentially extending $A(n)$ by the row a_{1n}^* and then by the column $(a_{1n}^* \ a_{nn})^*$ increases the rank each time. Thus, $\operatorname{rank} A = \operatorname{rank} A(n) + 2$, so n is Parter. Therefore, if n is downer or neutral, $E_A(0) \supseteq E'_{A(n)}(0)$. By definition, if n is downer, the containment is strict, and if n is neutral, the containment is actually equality.

Suppose n is Parter. Let X be the maximal subspace of $E'_{A(n)}(0)$ that is orthogonal to $(a_{1n}^* \ 0)^*$. Clearly, $X \subseteq E_A(0)$. Since $\dim X \geq m_{A(n)}(0) - 1 = m_A(0)$, we have $X = E_A(0)$. ∎

Proof of Theorem 2.1. Return to the calculation displayed in (2.1). Index i is null for A if and only if $E_A(\lambda) \subseteq E'_{A(i)}(\lambda)$. By the lemma, this is true if and only if i is Parter or neutral. ∎

3. Distinguishing Parter and neutral vertices. To distinguish between Parter and neutral vertices, then, we must look beyond the appropriate eigenspace of A itself. Our approach is to consider the secondary eigenspace, that of $A(i)$, associated with the same λ. We continue to write A as a block matrix as in (2.1). Again, we begin with some useful lemmas.

LEMMA 3.1. *If n is a null vertex, then n is neutral if and only if $E_{A(n)}(\lambda)$ is orthogonal to A_{1n}.*

Proof. By Lemma 2.2, $E_A(\lambda) \subseteq E'_{A(n)}(\lambda)$. In fact, $E_A(\lambda)$ is precisely the maximal subspace of $E'_{A(n)}(\lambda)$ that is orthogonal to $(a_{1n}^* \; 0)^*$. Thus, n is neutral if and only if $E_A(\lambda) = E'_{A(n)}(\lambda)$ if and only if $E_{A(n)}(\lambda)$ is orthogonal to a_{1n}. ∎

There is a particularly simple sufficient condition for orthogonality. We say that a subspace $X \subseteq \mathbb{C}^n$ is *combinatorially orthogonal* to a vector $y \in \mathbb{C}^n$ if $x_i\overline{y_i} = 0$ for every $x \in X, i = 1, \ldots, n$.

LEMMA 3.2. *Suppose that the graph of A is a tree and that n is a null vertex for some $\lambda \in \mathbb{R}$. The following statements are equivalent.*

1. *n is neutral.*
2. *All neighbors of n are null for $A(n)$.*
3. *$E_{A(n)}(\lambda)$ is orthogonal to a_{1n}.*
4. *$E_{A(n)}(\lambda)$ is combinatorially orthogonal to a_{1n}.*

Proof. $1 \Rightarrow 2$: If some neighbor of n were a nonzero vertex for $A(n)$, then that neighbor would be a downer vertex for its branch at n (Theorem 2.1). Then n would be Parter [JLDS].

$2 \Rightarrow 4$: The only nonzero entries in a_{1n} correspond to the neighbors of n. These neighbors are null vertices by assumption.

$4 \Rightarrow 3$ is obvious.

$3 \Rightarrow 1$ by Lemma 3.1. ∎

Example 3.3. The characterization of Parter vertices in terms of "downer branches" (specific to trees) is crucial to the proof of the lemma. In fact, if the graph of A is not a tree, then a neutral vertex i may be adjacent to a vertex j that is nonzero for $A(n)$. Consider

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

Vertex 3 is neutral for the eigenvalue 0, and vertices 1 and 2 are nonzero for $A(3)$. ∎

We may now state our main observation of this section, again focusing on trees. It identifies Parter vertices among null vertices by considering eigenspaces of $A(i)$.

THEOREM 3.4. *Let A be an Hermitian matrix whose graph is a tree, and let i be a null vertex for A. Then i is Parter if and only if there is a neighbor j that is nonzero for $A(i)$.*

Proof. This follows from the equivalence of 1 and 2 in Lemma 3.2, but we can also prove the result directly. The vertex i is Parter if and only if some neighbor of i is a downer vertex in its branch of $G(A) \setminus i$. If such a downer vertex exists, then it is nonzero for $A(i)$. If not, then every neighbor of i is null for $A(i)$. ∎

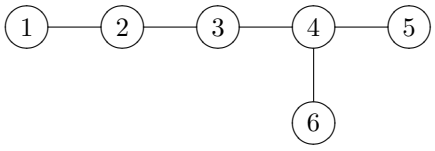
Our theorem has a surprising corollary.

COROLLARY 3.5. *Suppose that the graph of A is a tree. Every neighbor of a neutral vertex is a null vertex for A.*

Proof. By the theorem, if i is neutral, then every neighbor of i is null for $A(i)$. Because $E_A(\lambda) = E'_{A(i)}(\lambda)$, every vertex that is null for $A(i)$ is also null for A. ∎

The corollary implies, by Theorem 2.1, that every neighbor of a neutral vertex is either neutral or Parter. Thus, if a null vertex is not Parter, its neighbors constitute a natural place to look for Parter vertices. It can happen that all neighbors are again neutral, but, often, the neighbors include a Parter vertex.

Example 3.6. The converse of Corollary 3.5 is not true. It may happen that all neighbors of a null vertex are null without the vertex being neutral. Suppose an Hermitian matrix A with graph



satisfies $m_{A[\{1,2\}]}(\lambda) = m_{A[5]}(\lambda) = m_{A[6]}(\lambda) = 1$, $m_{A[3]}(\lambda) = 0$. Then vertex 4 is Parter because vertex 5 is a downer for its branch. A consequence of the discussion on paths following this example is that λ is not an eigenvalue of $A[\{1, 2, 3\}]$. Hence, $m_A(\lambda) = 1$. Now it is easy to check that vertex 2 is neutral and vertices 3 and 4 are Parter. ∎

Let A be an Hermitian matrix whose graph is a path, and let λ be an eigenvalue of A. For example, A could be an irreducible, tridiagonal Hermitian matrix. Using Theorem 2.1, we can locate the zeros in an eigenvector corresponding to λ. We begin by classifying the possible locations of Parter, neutral, and downer vertices. It is a well-known fact that deleting a pendant (i.e., degree 1) vertex from A causes the eigenvalue interlacing inequalities to be strict, and thus each pendant vertex is a downer vertex for λ. It follows that A has no neutral vertices, because if λ is an eigenvalue of $A(i)$, each neighbor of i is a pendant vertex in $G(A(i))$, and thus a downer vertex for its branch, forcing i to be Parter. For the same reason, if i and j are Parter vertices, then j is Parter for $A(i)$, and hence no two Parter vertices can be adjacent. The converse of these three observations is also true; specifically, if $i_1 \leq \cdots \leq i_k$ satisfy $i_1 \neq 1$, $i_k \neq n$, and $i_{j+1} - i_j > 1$ for $j = 1, \ldots, k-1$, then there exists an irreducible, tridiagonal Hermitian matrix for which λ is an eigenvalue and vertices $i_1, \ldots, i_k$ are precisely the Parter vertices. (Simply construct a B such that λ is an eigenvalue of each direct summand of $B(i_1, \ldots, i_k)$, and $B(i_1, \ldots, i_k)$ has no Parter vertices (trivial).) Furthermore, if λ is the rth largest (resp., smallest) eigenvalue of A, then λ can have at most $r-1$ Parter vertices, i.e., $k \leq r-1$. (To see this, iterate the interlacing inequalities to see that λ is the $(r-k)$th largest (resp., smallest) eigenvalue of $m_{A(\{i_1,\ldots,i_k\})}(\lambda)$.) Now, by Theorem 2.1, the constraints on $i_1, \ldots, i_k$ also characterize the locations of zeros in an eigenvector.

4. Implications. The observations made thus far show that there are simple but surprising links among the classification of vertices. These have some very strong implications that we explore here. First, we give two basic lemmas that hold independent of the graph and then consider implications via certain categories.

LEMMA 4.1. *Let A be an n-by-n Hermitian matrix. If i is neutral, then $j \neq i$ is downer for A if and only if j is downer for $A(i)$.*

Proof. If i is neutral, then $E_A(\lambda) = E'_{A(i)}(\lambda)$, which implies that j is nonzero for A if and only if j is nonzero for $A(i)$. ∎

LEMMA 4.2. *Let A be an n-by-n Hermitian matrix. If i is Parter and j is downer (for A and λ), then j is also downer for $A(i)$ and λ.*

Proof. Delete vertex i first and vertex j second. Then it is clear that $m_{A(\{i,j\})}(\lambda) \geq m_A(\lambda)$. Deleting the vertices in the opposite order gives $m_{A(\{i,j\})}(\lambda) \leq m_A(\lambda)$. Hence, $m_{A(\{i,j\})}(\lambda) = m_A(\lambda)$, which implies the result. ∎

4.1. Vertex classification. It is a goal for each graph G, $A \in \mathcal{H}(G)$, and identified $\lambda \in \mathbb{R}$, to be able to quickly classify each vertex w.r.t. Parter, neutral, or downer. In principle this could be done with prior results. Here, we mention some observations that assist in classification.

PROPOSITION 4.3. *Let A be an n-by-n Hermitian matrix. If $m_A(\lambda) = m$, then A has at least m downer vertices.*

Proof. Assume $m \geq 1$. Because $\dim E_A(\lambda) = m$, there is some vector in $E_A(\lambda)$ that has at least m nonzero entries. These entries identify downer vertices. ∎

PROPOSITION 4.4. *Suppose that the graph of A is connected. If $m_A(\lambda) = m \geq 1$, then A has at least $m+1$ downer vertices.*

Proof. By Proposition 4.3, A has at least m nonzero vertices. Suppose A has exactly m nonzero vertices. Then $E_A(\lambda)$ is spanned by vectors $e_{i_1}, \ldots, e_{i_m}$, where e_j is the jth standard basis vector for $\mathbb{C}^n$. Since $(A - \lambda I)e_j = 0$ implies the jth column of $A - \lambda I$ is zero, the graph of A is not connected. ∎

Example 4.5. A *star* is a graph that is a tree and has exactly one vertex of degree > 1. If the graph of A is the star on n vertices, and every diagonal entry of A is λ, then $m_A(\lambda) = n - 2$. Also, the central vertex is Parter, and every pendant vertex is a downer vertex, so A has exactly $m_A(\lambda) + 1$ downer vertices. Therefore, Proposition 4.4 is the strongest statement that can be made for all connected graphs. ∎

The following proposition is a restatement of Corollary 3.5.

PROPOSITION 4.6. *Suppose that the graph of A is a tree, and let i be a neutral vertex. Then every neighbor of i is either Parter or neutral for A.*

4.2. Classification of vertex pairs. We next turn to the classification of two vertices and, in particular, the possibilities for their status initially vs. sequentially. There are differences depending upon whether or not the two vertices are adjacent. We begin with another observation that is independent of the graph.

PROPOSITION 4.7. *Let A be an n-by-n Hermitian matrix, and let i and j be distinct indices. We have the following three statements.*

1. *If i and j are Parter, then $m_A(\lambda) - m_{A(\{i,j\})}(\lambda) \in \{-2, 0\}$.*
2. *If i and j are neutral, then $m_A(\lambda) - m_{A(\{i,j\})}(\lambda) \in \{-1, 0\}$.*
3. *If i is neutral and j is downer, then $m_A(\lambda) - m_{A(\{i,j\})}(\lambda) = 1$.*

Proof. 1. Clearly, if i and j are Parter vertices, then $-2 \leq m_A(\lambda) - m_{A(\{i,j\})}(\lambda) \leq 0$. Suppose that the difference is -1, for the sake of contradiction. Assuming w.l.o.g. that our eigenvalue λ equals 0 and that $i = n-1$ and $j = n$, we write

$$A = \left[\begin{array}{c|c|c} A(\{n-1,n\}) & a_{1,n-1} & a_{1,n} \\ \hline * & a_{n-1,n-1} & a_{n-1,n} \\ \hline * & * & a_{n,n} \end{array}\right],$$

where the entries marked $*$ are determined by the Hermicity of A. (Note that $A(\{n-1,n\})$ is our usual notation for the $(n-2)$-by-$(n-2)$ principal submatrix of A, that $a_{1,n-1}$ and $a_{1,n}$ are vectors of length $n-2$, and that all other entries are scalars.) By our assumption that $m_A(\lambda) - m_{A(\{n-1,n\})}(\lambda) = -1$, it follows that $n-1$ is neutral for $A(n)$ and that n is neutral for $A(n-1)$, and therefore $a_{1,n-1}$ and $a_{1,n}$ are linear combinations of the columns of $A(\{n-1,n\})$. Hence,

$$\operatorname{rank} A \leq \operatorname{rank} \begin{bmatrix} A(\{n-1,n\}) & a_{1,n-1} & a_{1,n} \end{bmatrix} + 2 = \operatorname{rank} A(\{n-1,n\}) + 2,$$

so that

$$m_A(0) = n - \operatorname{rank}(A) \geq (n-2) - \operatorname{rank} A(\{n-1,n\}) = m_{A(\{n-1,n\})}(0),$$

contradicting the assumption that $m_A(\lambda) - m_{A(\{n-1,n\})}(\lambda) = -1$.

2. By Lemma 4.1, if i and j are neutral, then j is Parter or neutral for $A(i)$.

3. By the same lemma, if i is neutral and j is downer, then j is downer for $A(i)$. □

COROLLARY 4.8. *Let A be an Hermitian matrix, and let i and j be distinct indices. If i is Parter and $m_A(\lambda) - m_{A(\{i,j\})}(\lambda) = -1$, then j is neutral for A.*

Proof. First, suppose that j is Parter. By Proposition 4.7, $m_A(\lambda) - m_{A(\{i,j\})}(\lambda) \neq -1$, a contradiction. Next, suppose that j is downer. Then $m_{A(\{i,j\})}(\lambda) \leq m_{A(j)}(\lambda) + 1 = m_A(\lambda)$, also a contradiction. □

PROPOSITION 4.9. *Suppose that the graph of A is a tree, and let i and j be neighbors. We have the following two statements.*

1. *If i and j are neutral, then $m_A(\lambda) - m_{A(\{i,j\})}(\lambda) = 0$.*
2. *If i and j are downer, then $m_A(\lambda) - m_{A(\{i,j\})}(\lambda) = 1$.*

Proof. 1. By Proposition 4.7, if i and j are neutral, then $m_A(\lambda) - m_{A(\{i,j\})}(\lambda) \in \{-1, 0\}$. Suppose $m_A(\lambda) - m_{A(\{i,j\})}(\lambda) = -1$. Then j is Parter in $A(i)$, so j is adjacent to a vertex k which is downer for $A(\{i,j\})$. But then k must also be a downer in $A(j)$ since i and j are adjacent. It follows that j is Parter for A—a contradiction.

2. If i and j are downer, then clearly $0 \leq m_A(\lambda) - m_{A(\{i,j\})}(\lambda) \leq 2$. Suppose that $m_A(\lambda) - m_{A(\{i,j\})}(\lambda) = 0$. Then j is Parter in $A(i)$, so j is adjacent to some vertex k which is downer for $A(\{i,j\})$. But since i and j are adjacent, k must also be downer for $A(j)$, which implies that j is Parter for A—a contradiction. Now suppose that $m_A(\lambda) - m_{A(\{i,j\})}(\lambda) = 2$. Then j is downer for its branch at i, which implies that i is Parter for A—a contradiction. □

Example 4.10. We will show that if i and j are not adjacent, then the conclusions of Proposition 4.9 may not hold.

First, observe that if an irreducible 2-by-2 Hermitian matrix has λ on its diagonal, then λ is not an eigenvalue.

Take $\lambda = 0$, and let

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

Check that $m_A(0) = 0$, and that the graph of A is a path. Removing either pendant vertex leaves a 2-by-2 Hermitian matrix with $\lambda = 0$ on its diagonal, so both pendant vertices are neutral. However, $m_{A(\{1,3\})}(0) = 1$, so claim 1 of Proposition 4.9 does not hold.

Still with $\lambda = 0$, take

$$B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

B has the same graph as A, but $m_B(0) = 1$. For the same reason as above, $m_{B(1)}(0) = m_{B(3)}(0) = 0$, so the pendant vertices are downer vertices. However, in contrast to claim 2 of Proposition 4.9, $m_B(0) - m_{B(\{1,3\})}(0) = 0$.

TABLE 4.1

i	j	$m_A(\lambda) - m_{A(\{i,j\})}(\lambda)$
Parter	Parter	-2, 0
Parter	Neutral	-1, 0
Parter	Downer	0
Neutral	Neutral	-1, 0
Neutral	Downer	1
Downer	Downer	0, 1, 2

Again with $\lambda = 0$, take

$$C = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Then $m_C(0) = 2$ and $m_{C(\{3,4\})}(0) = 0$, even though vertices 3 and 4 are downer.

We have seen that claim 2 of Proposition 4.9 need not hold for nonadjacent vertices, and, in fact, $m_A(\lambda) - m_{A(\{i,j\})}(\lambda)$ may take on either value 0 or 2. □

Using the results thus far, we are able to classify, for pairs of vertices, the joint or sequential effect upon multiplicity, given the individual effect of removal. It is of interest that certain possibilities cannot occur. Some of these are contained in results of this section thus far, and others are straightforward. We list the full classification without proof. Both the case of arbitrary and adjacent vertices are considered. In each case, a missing possibility provably cannot occur, and examples may be constructed for each listed possibility. For example, the last entry in the second table indicates that if a downer vertex is removed, an adjacent vertex that was initially a downer must then be neutral, and not Parter or downer (which can occur in the nonadjacent case).

Table 4.1 classifies the joint effect of removing two indices i and j in the following sense. Let Δ be an integer. If Δ is listed in some row of the table, then there exists an Hermitian matrix A and indices $i \neq j$ with given classifications such that $m_A(\lambda) - m_{A(\{i,j\})}(\lambda) = \Delta$. If Δ is missing from a row, then no matrix with such indices exists. Furthermore, for each listed Δ, an appropriate matrix A exists *whose graph is a tree*. Hence, the table would be identical if we restricted attention to matrices whose graphs are trees.

Table 4.2 concerns adjacent vertices. Specifically, if Δ is listed in some row of the table, then there exists an Hermitian matrix A *whose graph is a tree* and indices $i \neq j$ with given classifications *which are neighbors* such that $m_A(\lambda) - m_{A(\{i,j\})}(\lambda) = \Delta$. If Δ is missing from a row, then no matrix with such indices exists.

TABLE 4.2

i	j	$m_A(\lambda) - m_{A(\{i,j\})}(\lambda)$
Parter	Parter	-2, 0
Parter	Neutral	-1, 0
Parter	Downer	0
Neutral	Neutral	0
Neutral	Downer	not possible
Downer	Downer	1

The restriction to trees is important; without this restriction, the table would have been identical to the previous table. As before, the difference between trees and nontrees can be explained by the [JLDS] characterization of Parter vertices of trees.

4.3. Null subgraphs. This section is devoted to unexpected results about classification of entire subgraphs of a given graph.

Let G_1 be an induced subgraph of $G(A)$. If $x_i = 0$ for all $x \in E_A(\lambda)$, $i \in G_1$, then we say that G_1 is a *null subgraph* (for A and λ). Our first observation is a simple consequence of Theorem 2.1.

PROPOSITION 4.11. *G_1 is a null subgraph for A if and only if every vertex i of G_1 is Parter or neutral for A.*

If there is a sequence of vertices $i_1, \ldots, i_k$ such that i_1 is null for A and i_j is null for $A(\{i_1, \ldots, i_{j-1}\})$, $j = 2, \ldots, k$, then we say that $i_1, \ldots, i_k$ are *sequentially null*.

PROPOSITION 4.12. *Let A be an n-by-n Hermitian matrix, and suppose $i_1, \ldots, i_k$ are sequentially null. If λ is not an eigenvalue of some direct summand A_1 of $A(\{i_1, \ldots, i_k\})$, then $G(A_1)$ is a null subgraph for A.*

Proof. Clearly, every vertex in $G(A_1)$ is a null vertex for $A(\{i_1, \ldots, i_k\})$. For each j, the eigenspace of $A(\{i_1, \ldots, i_{j-1}\})$ is contained in the eigenspace of $A(\{i_1, \ldots, i_j\})$ (by Lemma 2.2), so null vertices of $A(\{i_1, \ldots, i_j\})$ are null vertices of $A(\{i_1, \ldots, i_{j-1}\})$. ∎

PROPOSITION 4.13. *Let A be an n-by-n Hermitian matrix, and suppose $i_1, \ldots, i_k$ are sequentially null. Identify some direct summand A_1 of $A(\{i_1, \ldots, i_k\})$. If $G(A_1)$ is a null subgraph for A, then $m_{A_1}(\lambda) \leq m_{A(\{i_1,\ldots,i_k\})}(\lambda) - m_A(\lambda)$.*

Proof. We have $E_A(\lambda) \subseteq E'_{A(\{i_1,\ldots,i_k\})}(\lambda)$. (Similar to the notation introduced before Lemma 2.2, $E'_{A(\{i_1,\ldots,i_k\})}(\lambda)$ is formed from $E_{A(\{i_1,\ldots,i_k\})}(\lambda)$ by inserting zeros into appropriate spaces.) If $A(\{i_1, \ldots, i_k\}) = A_1 \oplus A_2$, then $E_A(\lambda) \subseteq E'_{A_2}(\lambda)$. Now, $\dim E_{A_1}(\lambda) = \dim E_{A(\{i_1,\ldots,i_k\})}(\lambda) - \dim E_{A_2}(\lambda) \leq \dim E_{A(\{i_1,\ldots,i_k\})}(\lambda) - \dim E_A(\lambda)$. ∎

In the following proposition, T' is an induced subgraph of $G(A)$. The notation $A[T']$ denotes the principal submatrix of A lying in the rows and columns indexed by the vertices of T'.

PROPOSITION 4.14. *Suppose that the graph of A is a tree. Let i be Parter for A, and identify some branch T' at i for which $m_{A[T']}(\lambda) \geq 1$. If T' is a null subgraph for A, then every neighbor of i is Parter or neutral for A.*

Proof. If j is a neighbor of i, let T_j denote the branch of j at i.

In the notation of Lemma 2.2, we have $E_A(\lambda) \subseteq E'_{A(i)}(\lambda)$. Choose a basis B for $E'_{A(i)}(\lambda)$ in which the support of any basis vector is contained in a single branch T_j. Because $m_A(\lambda) - m_{A(i)}(\lambda) = -1$ and T' is a null subgraph, it follows that B contains exactly one vector x_1 whose support is T'. Furthermore, $B \setminus \{x_1\}$ is a basis for $E_A(\lambda)$.

Obviously, if there is a neighbor j such that no basis vector $x \in B \setminus \{x_1\}$ has support T_j, then j is a null vertex.

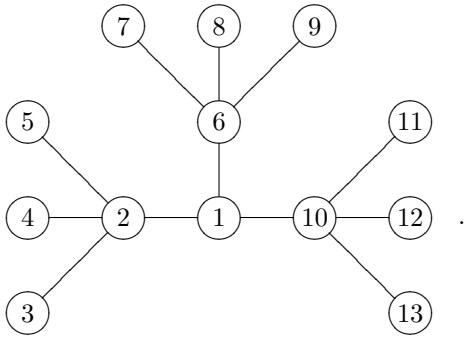
Now, let j be a neighbor of i, and suppose there exists an $x \in B \setminus \{x_1\}$ whose support is T_j. We have $(A - \lambda I_n)x = 0$, and $a_{ij}x_j = 0$ implies $x_j = 0$. ∎

5. Example. Results from the previous section can be used to classify vertices w.r.t. Parter, neutral, or downer with little knowledge of the numerical entries in a matrix. Of course, an understanding of the combinatorial structure of eigenspaces follows. Here we present an extended example.

Several results in the previous section concern the quantity $m_A(\lambda) - m_{A(\{i,j\})}(\lambda)$. Sometimes it is useful to think of extracting $A(\{i, j\})$ by first deleting row and column

i and then deleting row and column j. For example, if i is Parter for A and j is neutral for $A(i)$, then $m_A(\lambda) - m_{A(\{i,j\})}(\lambda) = -1$. In this case, we say that i and j are *sequentially Parter-neutral* (for A and λ). We may rephrase Corollary 4.8, "If i and j are sequentially Parter-neutral for A, then j is *originally* neutral for A."

Let $A = (a_{ij})$ be an Hermitian matrix with graph



.

Let $B = A[\{2,3,4,5\}]$, $C = A[\{6,7,8,9\}]$, and $D = A[\{10,11,12,13\}]$, the 4-by-4 principal submatrices of A lying in the indicated rows and columns. The graph of each of these principal submatrices is the star on four vertices. Let λ be a fixed real number, and suppose that $m_B(\lambda) = 0$, $m_C(\lambda) = 1$, and $m_D(\lambda) = 2$. Further, suppose that neither B nor C has Parter vertices. We will use this information to classify some of the vertices of A w.r.t. Parter, neutral, or downer and to completely classify the combinatorial structure of the eigenspace corresponding to λ.

Because $m_D(\lambda) = 2$, it follows that D has a Parter vertex and $a_{11,11} = a_{12,12} = a_{13,13} = \lambda$. Therefore, vertex 11 is a downer for its branch at vertex 10, so vertex 10 is Parter for A.

Similarly, vertex 6 is downer for C, so vertex 1 is Parter for A.

Because vertices 1 and 10 are sequentially Parter-Parter and $m_{A(\{1,10\})}(\lambda) = 4$, we conclude that $m_A(\lambda) = 2$.

By Proposition 4.12, the subgraph of $G(A)$ induced by vertices 2, 3, 4, 5 is a null subgraph, i.e., each vertex i, $i = 2,3,4,5$, is Parter or neutral for A.

Because vertices 1 and 2 are sequentially Parter-neutral, vertex 2 must be neutral for A by Corollary 4.8.

Because $m_A(\lambda) = 2$ and $m_{A[\{11,12,13\}]}(\lambda) = 3$, λ cannot be an eigenvalue of $A(\{10,11,12,13\})$ by the interlacing inequalities. Therefore, $m_A(\lambda) - m_{A(\{6,10\})}(\lambda) \leq -1$, so $m_A(\lambda) - m_{A(6)}(\lambda) \leq 0$. Vertex 6 is Parter or neutral for A. Similar arguments show that vertices 7, 8, and 9 are Parter or neutral.

By Proposition 4.4, A has at least three downer vertices, so vertices 11, 12, and 13 must be downers for A.

In summary, $m_A(\lambda) = 2$; vertices 1 and 10 are Parter; vertex 2 is neutral; vertices 11, 12, and 13 are downer; and each vertex i, $i = 3,4,5,6,7,8,9$, is either Parter or neutral. Therefore, $x_i = 0$ for all $x \in E_A(\lambda)$, $i \neq 11,12,13$, and there is some eigenvector which is nonzero in coordinates 11, 12, and 13.

Remark 5.1. Once we know that vertex 2 is neutral, Proposition 4.6 implies that vertices 3, 4, and 5 are null vertices, which agrees with our findings.

Remark 5.2. Once we know that vertices 6, 7, 8, and 9 are null vertices, Proposition 4.14 implies that vertices 2 and 10 are also null vertices, which agrees with our findings.

Remark 5.3. The constraints on A were insufficient to classify every vertex as Parter, neutral, or downer. For example, there is a matrix that satisfies the constraints on A such that vertex 3 is Parter but vertex 4 is neutral, and vertex 7 is Parter but vertex 8 is neutral. It is also possible to show that vertex 6 may be either neutral or Parter. However, if vertex 6 is Parter, then vertices 7, 8, and 9 must be neutral, because then vertices 6 and i, $i = 7, 8, 9$, would be sequentially Parter-neutral.

Acknowledgment. The authors wish to thank Lon Mitchell for his suggestion to pursue the results herein because of graph theoretic interest.

REFERENCES

[HJ] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985, 1990.

[JLD99] C. R. Johnson and A. Leal Duarte, *The maximum multiplicity of an eigenvalue in a matrix whose graph is a tree*, Linear Multilinear Algebra, 46 (1999), pp. 139–144.

[JLD02] C. R. Johnson and A. Leal Duarte, *On the possible multiplicities of the eigenvalues of a Hermitian matrix whose graph is a tree*, Linear Algebra Appl., 348 (2002), pp. 7–21.

[JLDS] C. R. Johnson, A. Leal Duarte, and C. M. Saiago, *The Parter-Wiener theorem: Refinement and generalization*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 352–361.

[JLD+] C. R. Johnson, A. Leal Duarte, C. M. Saiago, B. D. Sutton, and A. J. Witt, *On the relative position of multiple eigenvalues in the spectrum of an Hermitian matrix with a given graph*, Linear Algebra Appl., 363 (2003), pp. 147–159.

[P] S. Parter, *On the eigenvalues and eigenvectors of a class of matrices*, J. Soc. Indust. Appl. Math., 8 (1960), pp. 376–388.

[W] G. Wiener, *Spectral multiplicity and splitting results for a class of qualitative matrices*, Linear Algebra Appl., 61 (1984), pp. 15–29.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 2, pp. 400–414

SOLVING POLYNOMIALS WITH SMALL LEADING COEFFICIENTS*

GUDBJÖRN F. JÓNSSON† AND STEPHEN VAVASIS‡

Abstract. We explore the computation of roots of polynomials via eigenvalue problems. In particular, we look at the case when the leading coefficient is relatively very small. We argue that the companion matrix algorithm (used, for instance, by the Matlab `roots` function) is inaccurate in this case. The accuracy problem is addressed by using matrix pencils instead. This improvement can be predicted from the backward error bound of Edelman and Murakami (for companion matrices) versus the bound of Van Dooren and Dewilde (for pencils). We then show how to extend the accurate algorithm to Bézier polynomials and present computational experiments.

Key words. polynomial roots, stability, condition number, generalized eigenvalue, matrix pencil, Bézier polynomial

AMS subject classifications. 12Y05, 15A22, 65F15, 65H05

DOI. 10.1137/S0895479899365720

1. Introduction.

Computing the roots of a univariate polynomial is a fundamental problem that arises in many applications. The focus of this paper is on polynomials where the leading coefficient is much smaller than some of the other coefficients. Such polynomials occur frequently in geometric applications like mesh generation and graphics. The reason is that the user of geometric applications often works witha fixed toolbox of geometric primitives, e.g., cubic splines. The application might store a user's linear or quadratic polynomial as a cubic with leading coefficients of zeros. Once transformations (such as translations and rotations) are applied, the leading coefficient, which was formerly zero, could become small but nonzero in these cases. An example of such a geometric application is the QMG mesh generator [9]. Even in an implementation of the quadratic formula to solve a real polynomial $ax^2 + bx + c = 0$, there are numerical difficulties when b is much bigger than the other two coefficients. This problem and its solution are discussed in many textbooks on numerical analysis (see, e.g., Example 1.10 of [6]).

One way of numerically computing the roots of a polynomial is to form its companion matrix and compute the eigenvalues. This is, for example, how the Matlab function `roots` works [10]. There exist quality algorithms for computing eigenvalues, so `roots` should give accurate solutions as long as the following two conditions are met. First, the problem has to be well conditioned to begin with. A root ξ of a polynomial $q(x)$ is *well conditioned* if the quantity $\kappa_2(\xi, q)$, defined by (2.2) below and to be discussed later, is not too large. Second, the translation from a polynomial to an eigenvalue problem should not cause the conditioning of the problem to become much worse. Our focus is on the latter issue.

Let p be a polynomial,

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0,$$

*Received by the editors December 10, 1999; accepted for publication in revised form) August 26, 2003; published electronically November 17, 2004.
http://www.siam.org/journals/simax/26-2/36572.html
†Decode Genetics, Sturlugata 8, Reykjavik, Iceland (gudbjorn.jonsson@decode.is). Research supported in part by NSF grants CCR-9619489 and EIA-9726388. This research also supported in part by NSF through grant DMS-9505155 and ONR through grant N00014-96-1-0050.
‡Department of Computer Science, Cornell University, Ithaca, NY 14853 (vavasis@cs.cornell.edu).

and C be its companion matrix,

$$C = \begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -\frac{a_0}{a_n} & \cdots & \cdots & -\frac{a_{n-1}}{a_n} \end{bmatrix}. \tag{1.1}$$

Suppose the eigenvalues of C are computed using a backward stable eigensolver, so that they are the exact eigenvalues of $C + E$, where E is a matrix with small entries. The computed eigenvalues are also roots of a perturbed polynomial $\tilde{p}$ with coefficients $\tilde{a}_j = a_j + e_j$. Edelman and Murakami [2] show that to first order

$$e_{j-1} \simeq \sum_{m=0}^{j-1} a_m \sum_{i=j+1}^{n} E_{i+m-j,i} - \sum_{m=j}^{n} a_m \sum_{i=1}^{j} E_{i+m-j,i},$$

taking into account that our companion matrix is the transpose of theirs. Note that the leading coefficient a_n is not perturbed ($e_n = 0$). Also, note that this formula is still correct even though we do not assume the normalization $a_n = 1$ used in [2].

In `roots` the eigenvalue problem $C\mathbf{x} = \lambda\mathbf{x}$ is solved using the QR-algorithm (see, for example, section 7.5 of [5]), and it can be shown that

$$\|E\| < k_1 \|C\| \epsilon_{\text{mach}},$$

where ϵ_{mach} is machine epsilon, $\|\cdot\| = \|\cdot\|_{\text{F}}$ is the Fröbenius norm, and k_1 depends only on n. (Actually k_1 also depends on the number of QR-steps, but in Matlab's implementation the algorithm is deemed to have failed to converge if this number exceeds $3n$. For a detailed backward error analysis of the QR-algorithm see [12].) Let $\mathbf{a} = [a_0, \ldots, a_n]$ and $\tilde{\mathbf{a}} = [\tilde{a}_0, \ldots, \tilde{a}_n]$. Now, $\|C\| \simeq |a_{\max}/a_n|$, where $a_{\max}$ is the largest coefficient, i.e., $|a_{\max}| = \max |a_j|$. So we get a backward error bound,

$$\|\tilde{\mathbf{a}} - \mathbf{a}\| < k_2 \left| \frac{a_{\max}}{a_n} \right| \cdot \|\mathbf{a}\| \cdot \epsilon_{\text{mach}} + O(\epsilon_{\text{mach}}^2). \tag{1.2}$$

Here $\|\cdot\| = \|\cdot\|_2$. (In this paper all vector norms are 2-norms and matrix norms in this section are Fröbenius norms. Also, from now on the error bounds are only written to first order in ϵ_{mach}, i.e., the $O(\epsilon_{\text{mach}}^2)$ term is omitted.) This bound is not so good when $a_{\max}$ is much bigger than the leading coefficient a_n, the case we will refer to by saying that *p has a small leading coefficient*.

In a paper on matrix polynomials [14], Van Dooren and Dewilde present an analysis of a different algorithm. If their result is written for ordinary polynomials, it goes as follows. Consider the generalized eigenvalue problem $A\mathbf{x} = \lambda B\mathbf{x}$, with

$$A - \lambda B = \begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -a_0 & \cdots & \cdots & -a_{n-1} \end{bmatrix} - \lambda \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & a_n \end{bmatrix}. \tag{1.3}$$

If this is solved using the QZ-algorithm (section 7.7 of [5]), the computed eigenvalues are exact for a perturbed matrix pencil

$$(A + E) - \lambda(B + F), \tag{1.4}$$

with

$$\|E\| < k_a \|A\| \epsilon_{\text{mach}}, \qquad \|F\| < k_b \|B\| \epsilon_{\text{mach}},$$

where k_a, k_b depend only on n.

Let $\mathbf{a} = [a_0, \ldots, a_n]$, as before, and assume that the coefficients have been scaled so that $\|\mathbf{a}\| = 1$. In a clever way Van Dooren and Dewilde use row and column operations to get the perturbed pencil (1.4) back into the same form as the original pencil (1.3), thus showing that the computed roots are exact for a polynomial $\tilde{p}$ with

$$\|\tilde{\mathbf{a}} - \mathbf{a}\| < k_3 \epsilon_{\text{mach}} = k_3 \|\mathbf{a}\| \epsilon_{\text{mach}}, \tag{1.5}$$

where k_3 depends only on n.

One would expect solving $C\mathbf{x} = \lambda \mathbf{x}$ and solving $A\mathbf{x} = \lambda B\mathbf{x}$ to give similar results, but clearly the error bound (1.5) is better than (1.2). The difference is the factor $|a_{\max}/a_n|$, which comes from the norm of the companion matrix C. The advantage of the matrix pencil is that we can normalize the entries of the matrices A and B, so that we get bounds on the matrix norms independent of the coefficients a_j. For the companion matrix we do not have this freedom.

We must also point out that there is a difference in the way the two methods are analyzed. Edelman and Murakami [2] fix the leading coefficient, while Van Dooren and Dewilde [14] allow all the coefficients to be perturbed. In section 2 we verify by numerical examples that both the normalization $\|\mathbf{a}\| = 1$ and perturbing the leading coefficient are needed for the strong backward error bound (1.5) to hold.

We then compare the accuracy of the roots computed by the two methods. The results are given in relation to the condition number of each root, the hope being that the forward error is of the order of condition number times machine precision, or smaller. If we use the pencil (1.3), this is indeed the case. For polynomials with a small leading coefficient and roots of order 1 in magnitude or smaller, this method does better than `roots`. For other classes of problems, `roots` sometimes gives better answers than the pencil algorithm; see further remarks in section 4.

In section 3 we turn our attention towards Bézier polynomials, i.e., polynomials arising in computations with Bézier curves. We propose two generalized eigenvalue approaches for computing their roots, and give a backward error bound for one of them. Numerical experiments are done to reveal the benefit of the generalized eigenvalue approach over using the companion matrix together with a change of variables.

There certainly are other ways of dealing with a small leading coefficient. One way is to drop the leading term, if that causes a smaller error than we would get if we kept it. Another way is to use linear fractional transformations. We return to these approaches in the next section.

2. Numerical experiments. To test the two methods we need polynomials whose coefficients vary greatly in size. We generate random test polynomials as follows. First we form a random polynomial of degree 8 with coefficients

$$(\alpha + i\beta) \cdot 10^{\gamma},$$

where α and β are chosen uniformly from $[-1, 1]$ and γ is chosen uniformly from $[-10, 10]$. To get a small leading coefficient we fix it at 10^{-10}. (This is slightly modified from the random test polynomials used in [13].) We then impose a double root at $1/2$ by multiplying this polynomial with $(z - \frac{1}{2})^2$, thus giving a test polynomial

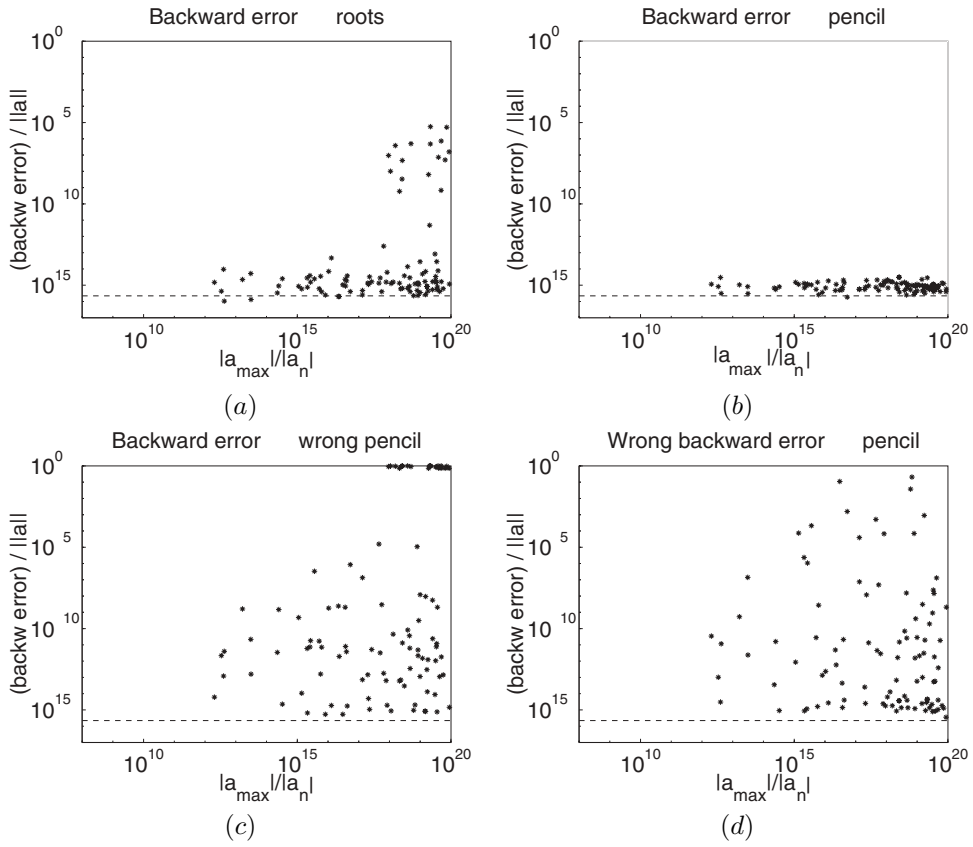


FIG. 2.1. *Backward error:* (a) `roots`, (b) *and* (d) *normalized pencil* ($\|\mathbf{a}\| = 1$), (c) *pencil with* $a_n = 1$. *All minimal errors except* (d) *where* a_n *is fixed. Dashed line is at machine epsilon.*

of degree 10. We plot the forward error versus the condition number of the root and the double root is added to ensure that there is an ill-conditioned root.

Given computed roots, $\hat{z}_1, \ldots, \hat{z}_n$, let

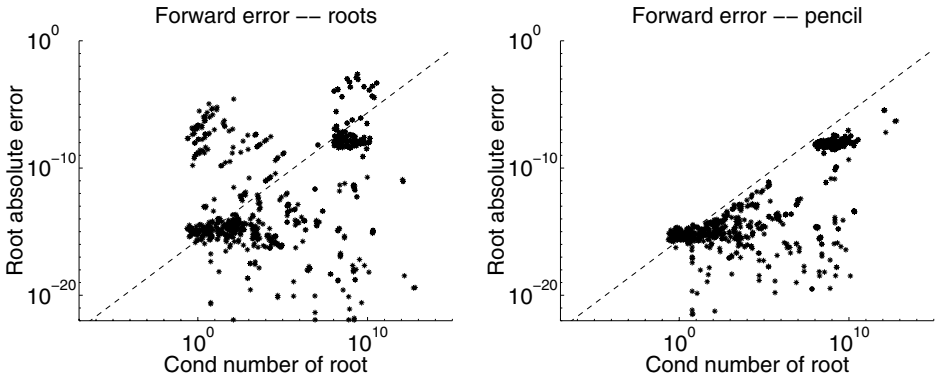
$$\hat{p}(z) = (z - \hat{z}_1) \cdots (z - \hat{z}_n).$$

We compute the coefficients, $\hat{a}_0, \ldots, \hat{a}_n$, of this polynomial using 40-decimal-digit precision (via Matlab's `vpa` function). If we allow all the coefficients of p to be perturbed, the perturbation giving $\hat{z}_1, \ldots, \hat{z}_n$ as exact roots is not unique, since multiplying $\hat{p}$ by a scalar does not change its roots. Unless otherwise stated, we will compute the backward error that is minimal in a least squares sense,

$$\min_{\tau} \|\tau\hat{\mathbf{a}} - \mathbf{a}\|. \tag{2.1}$$

The minimum is obtained at $\tau = (\hat{\mathbf{a}}^H\mathbf{a})/(\hat{\mathbf{a}}^H\hat{\mathbf{a}})$.

Figure 2.1 shows the relative normwise backward error as the roots and the error are computed in four different ways (all using the same 100 random polynomials). "Normwise" means that we measure the norm $\|\mathbf{a} - \tau\hat{\mathbf{a}}\|$ as opposed to componentwise errors in individual coefficients. We return to componentwise bounds in section 4.



(a) (b)

FIG. 2.2. *Forward error:* (a) `roots`, (b) *normalized pencil. Error shown only for roots with modulus* < 10. *Dashed line is the condition number times machine epsilon.*

In (a) we used Matlab's `roots` and the error is computed using (2.1). Even if all the coefficients are perturbed and we take the smallest perturbation, we get an error substantially bigger than machine epsilon. So it is not possible to get an error bound as strong as (1.5) for this method. In fact, this is only slightly better than we would get holding a_n fixed ($\tau = a_n/\hat{a}_n$).

The plot (b) shows the backward error when the roots are computed by solving the generalized eigenvalue problem $A\mathbf{x} = \lambda B\mathbf{x}$, the coefficients are normalized so that $\|\mathbf{a}\| = 1$, and (2.1) is used to compute the backward error. We see that the pencil algorithm gives much better results than the companion matrix algorithm for our test problems. To see how important the normalization $\|\mathbf{a}\| = 1$ is, we modified the pencil algorithm in (c) by using the normalization $a_n = 1$ instead. (Basically we are using the pencil $C - \lambda I$.) And in (d) we used the same algorithm as in (b) but computed the backward error using $\tau = a_n/\hat{a}_n$ to see what happened if we insisted on not perturbing a_n. We see that both the normalization $\|\mathbf{a}\| = 1$ and perturbing all the coefficients are needed for (1.5) to hold.

Next we turn to forward error, i.e., accuracy of the computed roots. In what follows, $z_1, \ldots, z_n$ will denote the "exact" roots, which we find by simplifying in 40-decimal-digit arithmetic the result of Matlab's `solve` function. The roots computed in regular (double) precision are denoted by $\hat{z}_1, \ldots, \hat{z}_n$.

Figure 2.2 shows the error $|\hat{z}_j - z_j|$ for the same set of 100 random polynomials used for the backward error estimates. We choose to plot the accuracy of each computed root as a function of its condition number. Let

$$q(z) = z^n + c_{n-1}z^{n-1} + \cdots + c_1 z + c_0$$

be a monic polynomial. Toh and Trefethen [13] derive a formula for the condition number of a root ξ of q, which for normwise perturbations becomes

$$\kappa_1(\xi, q) = \| [c_0, \ldots, c_{n-1}] \| \frac{\| [1, \xi, \ldots, \xi^{n-1}] \|}{|q'(\xi)|}.$$

By "normwise perturbations" we mean that this formula bounds how much ξ will change if the coefficient vector $\mathbf{c} = [c_0, \ldots, c_{n-1}]$ is perturbed by a vector $\mathbf{u}$ such that

$\|\mathbf{u}\| \ll \|\mathbf{c}\|$. In section 4 we consider componentwise perturbations in which $|u_i| \ll |c_i|$ individually for each i. Toh and Trefethen did not perturb the leading coefficient, but if we allow perturbation of all the coefficients and modify the derivation in [13] accordingly, we get

$$\kappa_2(\xi, p) = \|\mathbf{a}\| \frac{\| [1, \xi, \ldots, \xi^n] \|}{|p'(\xi)|}, \tag{2.2}$$

as the condition number of a simple root ξ of p. This is the formula we use.

In geometric computing, we usually only need the roots in the interval $[0, 1]$, but we want to be sure we have all of them with high accuracy. Since this is the intended application, our goal is to get the roots with small modulus as accurate as possible, while the accuracy of the big ones is less important. We return to the matter of computing the large roots in section 4. Therefore, the plots in Figure 2.2 include only roots satisfying $|z| < 10$. It is for these roots that the method using $A\mathbf{x} = \lambda B\mathbf{x}$ does well compared to `roots`. We see in plot (b) that the roots computed using the pencil algorithm have errors $< \kappa_2 \epsilon_{\text{mach}}$, while this is not true for the companion matrix algorithm seen in plot (a).

We conclude this section by looking at two other possible approaches to handling a small leading coefficient. As mentioned in the introduction, another approach is to drop leading coefficients that are too small. For example, suppose the size of the leading coefficient is $\delta\|\mathbf{a}\|$. Then the error in the roots caused by the $a_{\max}/a_n$ factor in (1.2), assuming the leading coefficient is not dropped, is on the order of $\epsilon_{\text{mach}}/\delta$. On the other hand, the error caused from the change to the polynomial (which therefore changes the roots) for dropping the leading coefficient is on the order of δ. This suggests that the best strategy is to drop the leading coefficient (and subsequent leading coefficients, if they continue to be small) if $\delta \leq \sqrt{\epsilon_{\text{mach}}}$. The worst error in this case is expected to be about $\sqrt{\epsilon_{\text{mach}}}$. We ran our tests on this method; the results are illustrated in Figure 2.3(a) and (b) (backward and forward errors). As before, only forward error of roots with modulus less than 10 are depicted. As is evident from the figure (and is expected from the preceding explanation), this method is not as accurate as the pencil method. We also tried augmenting this method by computing a Newton step to improve each root. (The rationale is that the worst case error is $\sqrt{\epsilon_{\text{mach}}}$, so we could, in principle, get a nearly exact root with a single step of Newton to double the number of digits.) The results of this augmentation (not shown) are that the forward errors are often improved, but the backward errors often get worse with this modification.

Another potential approach for handling a small leading coefficient is to change variables by a fractional linear transformation (FLT), and then transform the computed roots back after the solution. This approach is difficult to implement because the best FLT for the data at hand is problem-dependent and is not easy to determine in advance. Therefore, a rootfinder that adopted this approach would probably have to try several FLTs chosen at random and use a cutoff measure to determine which FLT was best. We tried solving for the roots of the 100 polynomials using the fixed FLT $t = 1/(z-1)$ which maps the very large root or roots of the original polynomial (caused by the small leading coefficient) to roots near 1. This algorithm performed acceptably as shown in Figure 2.3(c) and (d), but still worse than the pencil algorithm. Furthermore, as already mentioned, the use of a single fixed FLT is not recommended as a general solution to the problem of small leading coefficients.

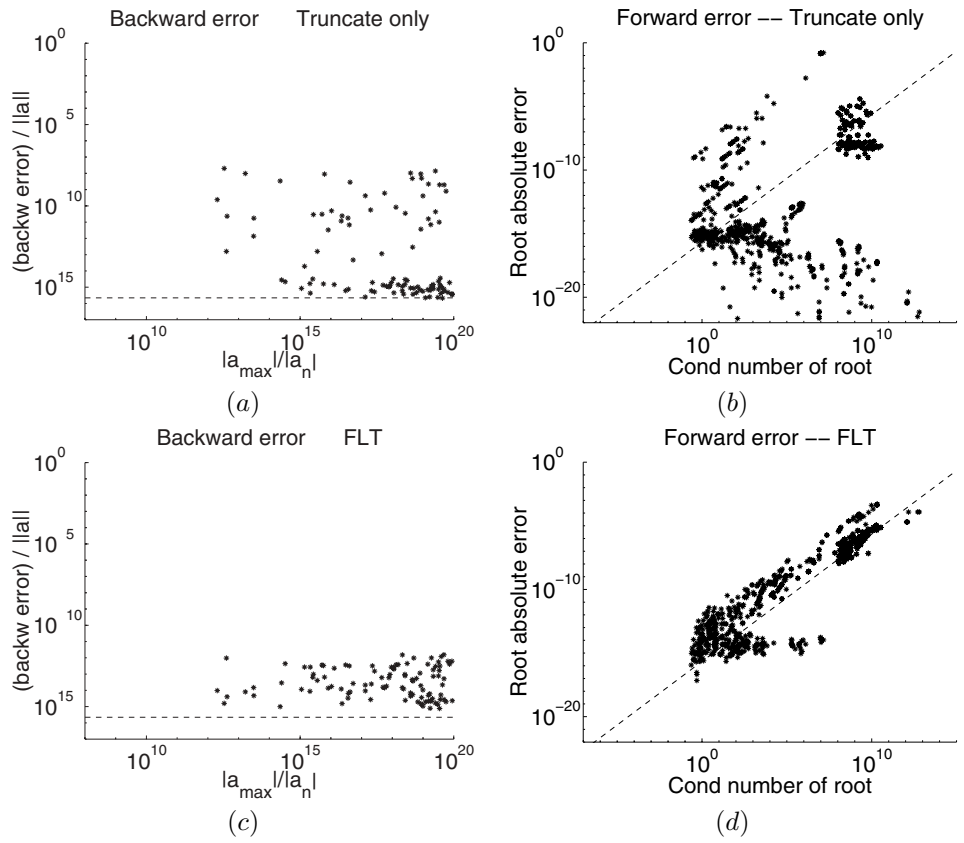


FIG. 2.3. *The method of truncating small leading terms (backward and forward errors) are depicted in* (a) *and* (b)*, while the method of a fractional linear transformation is depicted in* (c) *and* (d).

3. Bézier polynomials. Bézier curves are widely used in geometric computing [3]. Given *control points* $\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_n$, the Bézier curve $\mathbf{b}(t)$ is defined for $t \in [0,1]$ by taking repeated convex combinations

$$\begin{aligned}
\mathbf{b}_j^1 &= (1-t)\mathbf{b}_j + t\mathbf{b}_{j+1}, \quad j = 0, \ldots, n-1, \\
\mathbf{b}_j^2 &= (1-t)\mathbf{b}_j^1 + t\mathbf{b}_{j+1}^1, \quad j = 0, \ldots, n-2, \\
&\vdots \\
\mathbf{b}_0^n &= (1-t)\mathbf{b}_0^{n-1} + t\mathbf{b}_1^{n-1}
\end{aligned}$$

and setting $\mathbf{b}(t) = \mathbf{b}_0^n$. An equivalent definition is given by

$$\mathbf{b}(t) = \sum_{j=0}^{n} \mathbf{b}_j \binom{n}{j} (1-t)^{n-j} t^j.$$

(The latter is called the Bernstein form of the Bézier curve.) Figure 3.1 shows an example with three control points.

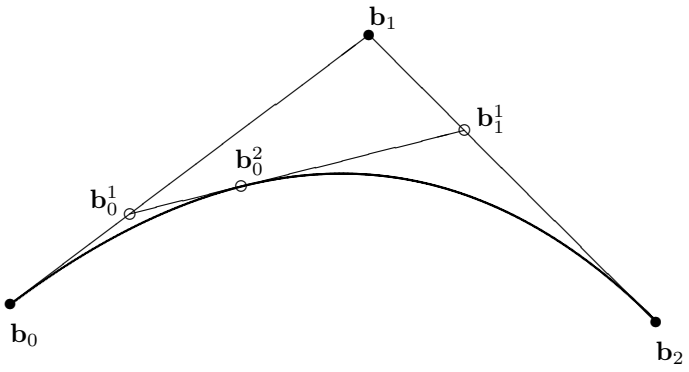


FIG. 3.1. *An example of a Bézier curve (with the points* $\mathbf{b}_j^r$ *for* $t = 1/3$*).*

If we wanted to know where a Bézier curve intersects one of the coordinate axes, we would have to solve a polynomial equation

$$p_B(t) = \sum_{j=0}^{n} b_j \binom{n}{j} (1-t)^{n-j} t^j = 0, \tag{3.1}$$

where the b_j are now real scalars. In general, the problem of finding the intersection of a Bézier curve and an arbitrary line can be written as this kind of equation.

Let $a_j = \binom{n}{j} b_j$. One way of solving $p_B(t) = 0$ is to solve

$$q_B(z) = a_n z^n + \cdots + a_1 z + a_0 = 0$$

using `roots` and then compute fractional transformation $t = z/(1+z)$. This approach may run into numerical difficulties as above, namely, the method will work poorly if a_n is much smaller than the other coefficients. A better approach is to run the QZ algorithm on the Van Dooren–Dewilde pencil given by (1.3) for q_B followed by the fractional transformation. At first glance, it seems that this method could run into difficulties if one or more of the roots of q_B is close to -1 since the denominator in the equation $t := z/(1+z)$ will blow up. In fact, the following analysis shows that this algorithm is backward stable for finding roots of a Bézier polynomial.

Let $\hat{z}_1, \ldots, \hat{z}_n$ be the computed roots of q_B using the pencil algorithm. This analysis will assume that no $\hat{z}_i$ is exactly equal to -1, although the analysis could be extended to that case as well (i.e., there is no assumption of a lower bound on the distance to -1). Let $\hat{q}_B(z) = (z - \hat{z}_1) \cdots (z - \hat{z}_n)$. The backward error bound of Van Dooren and Dewilde implies that there exists a τ_1 such that

$$\|q_B - \tau_1 \hat{q}_B\| \le C_n \epsilon_{\text{mach}} \|q_B\|. \tag{3.2}$$

Here, we regard q_B and $\hat{q}_B$ as vectors with $n+1$ entries, the constant coefficient being the first entry and the leading coefficient being the last.

Let $T_1, \ldots, T_n$ be the computed values of $\hat{z}_1/(\hat{z}_1+1), \ldots, \hat{z}_n/(\hat{z}_n+1)$. We adopt the usual model of floating-point arithmetic, namely, that every operation $+, -, *, /$ on floating point numbers x, y yields the correct answer multiplied by a factor $(1+\delta)$, where $|\delta| \le \epsilon_{\text{mach}}$, machine epsilon. Assuming the model is valid for real numbers,

Higham [7] shows the same bounds apply to complex arithmetic except with a slightly larger value of ϵ_{mach} and with the proviso that δ may now be complex. For the rest of this section, we rely on this result and we redefine ϵ_{mach} to be machine epsilon for complex floating-point arithmetic (i.e., we multiply the original machine epsilon by a small constant factor). Refer also to Exercise 1.12 of Demmel [1].

Thus, we can say that

$$T_i = \frac{(1+\eta_i)\hat{z}_i}{(1+\hat{\eta}_i)(\hat{z}_i+1)},$$

where $\eta_i, \hat{\eta}_i$ are two complex numbers satisfying $|\eta_i|, |\hat{\eta}_i| \leq \epsilon_{\text{mach}}$ and accounting for the roundoff in the addition in the denominator and quotient. Since $(1+\eta_i)/(1+\hat{\eta}_i) = 1+\eta_i-\hat{\eta}_i+O(\epsilon_{\text{mach}}^2)$, we can let $\delta_i = \eta_i - \hat{\eta}_i$ and drop the high order term to simplify:

$$T_i = \frac{(1+\delta_i)\hat{z}_i}{\hat{z}_i+1},$$

where $|\delta_i| \leq 2\epsilon_{\text{mach}}$.

Next, let $P_B(t)$ be the Bézier polynomial whose roots are $T_1, \ldots, T_n$. The goal for the backward error analysis is to show that the control points of P_B, say $B_0, \ldots, B_n$, are normwise not too far away from $b_0, \ldots, b_n$, i.e., to show that there exists a τ_2 such that

$$\|(b_0, \ldots, b_n) - \tau_2(B_0, \ldots, B_n)\| \leq O(\epsilon_{\text{mach}}) \cdot \|(b_0, \ldots, b_n)\|. \tag{3.3}$$

Define $A_i = n!B_i/(i!(n-i)!)$. Let $Q_B(z) = A_n z^n + \cdots + A_0$. Then the above problem is equivalent to showing that there exists τ_2 such that

$$\|q_B - \tau_2 Q_B\| \leq O(\epsilon_{\text{mach}}) \cdot \|q_B\|, \tag{3.4}$$

where the constant in the O-notation of (3.4) differs from the constant in (3.3) by a multiplicative factor at most $n!/((n/2)!(n/2)!)$.

We will prove (3.4) in two steps: First, (3.2) gives the distance from q_B to $\hat{q}_B$. Second, we derive an inequality for the distance from $\hat{q}_B$ to Q_B. Let $Z_1, \ldots, Z_n$ be the result of applying inverse transformation $Z = t/(1-t)$ to $T_1, \ldots, T_n$. Clearly $Z_1, \ldots, Z_n$ are the roots of $Q_B(z)$. Assume $Q_B(z) = (z - Z_1) \cdots (z - Z_n)$ to fix the scaling of Q_B. An initial approach for bounding $\hat{q}_B - Q_B$ would be to argue that for each i, $\hat{z}_i$ is close to Z_i in a relative sense. In fact, this argument is not valid as Z_i could be relatively very distant from $\hat{z}_i$ when the latter is very large. The correct argument does not rely on relative closeness.

Regard Q_B and $\hat{q}_B$ as vectors in $\mathbf{C}^{n+1}$ with constant coefficient written as the first entry and leading coefficient as the last. Observe that Q_B can be obtained by the following matrix computations. Start with the vector [1] in $\mathbf{C}^1$, which represents the constant zero-degree polynomial 1. Next, apply the 2×1 matrix $H_1 = (-Z_1; 1)$ to this vector to yield a 2-vector representing the linear polynomial $z - Z_1$. Next, apply the 3×2 matrix

$$H_2 = \begin{pmatrix} -Z_2 & 0 \\ 1 & -Z_2 \\ 0 & 1 \end{pmatrix}$$

to $H_1 \cdot [1]$ to obtain a quadratic polynomial, and so on. Thus $Q_B = H_n \cdots H_1$, where we omit the trailing factor [1] by identifying $\mathbf{C}^n$ with $\mathbf{C}^{n \times 1}$. Similarly, $q_B = \hat{H}_n \cdots \hat{H}_1$,

where $\hat{H}_i$ is an $(i+1) \times i$ bidiagonal Toeplitz matrix with $-\hat{z}_i$ on the main diagonal instead of $-Z_i$.

Observe that the maximum and minimum singular values of H_i can be written in closed form (since every eigenvalue and eigenvector of $H_i^* H_i$ can be written in terms of trig functions):

$$\sigma_{\max}(H_i) = (|Z_i|^2 + 2|Z_i| \cos(\pi/(i+1)) + 1)^{1/2} \tag{3.5}$$

and

$$\sigma_{\min}(H_i) = (|Z_i|^2 - 2|Z_i| \cos(\pi/(i+1)) + 1)^{1/2}. \tag{3.6}$$

From these two equations, we can easily obtain two bounds,

$$\sigma_{\max}(H_i) \leq |Z_i| + 1, \tag{3.7}$$

obtained from (3.5) using the estimate $\cos(\cdot) \leq 1$, and

$$\sigma_{\min}(H_i) \geq \max(\sin(\pi/(i+1)), |Z_i| - 1), \tag{3.8}$$

where the first term in the max comes from minimizing the right-hand side of (3.6) over all choices of $|Z_i|$ and the second term comes from the estimate $\cos(\cdot) \leq 1$. Similar equations and similar bounds apply to $\hat{H}_i$ with $\hat{z}_i$ replacing Z_i.

We claim that for each i, there exists a μ_i such that

$$\|\hat{H}_i - \mu_i H_i\| \leq 3\epsilon_{\text{mach}} \cdot (1 + |\hat{z}_i|). \tag{3.9}$$

In (3.9) and for the rest of the section, we use the matrix 2-norm, i.e., the largest singular value. Recall that $Z_i = T_i/(1 - T_i)$ and $T_i = (1 + \delta_i)\hat{z}_i/(\hat{z}_i + 1)$. (We assume $T_i \neq 1$, i.e., Z_i is finite, but the analysis can be extended to the case $T_i = 1$ as well.) Combining,

$$Z_i = \frac{(1 + \delta_i)\hat{z}_i}{1 - \delta_i \hat{z}_i}. \tag{3.10}$$

There are now three cases. If $\hat{z}_i = 0$ then $Z_i = 0$ also (and conversely) hence in this case, $H_i = \hat{H}_i$, satisfying (3.9) for $\mu_i = 1$. The other two cases are that $|\hat{z}_i| \leq 2$ or $|\hat{z}_i| > 2$. If $|\hat{z}_i| \leq 2$, start from (3.10) and drop high-order terms to obtain

$$Z_i = (1 + \delta_i)(1 + \delta_i \hat{z}_i)\hat{z}_i = (1 + \delta_i + \delta_i \hat{z}_i)\hat{z}_i = \hat{z}_i + (1 + \hat{z}_i)\delta_i;$$

hence $|Z_i - \hat{z}_i| \leq 2\epsilon_{\text{mach}}(1 + |\hat{z}_i|)$ (since $|\delta_i| \leq 2\epsilon_{\text{mach}}$). Since the off-diagonal entries of $\hat{H}_i - H_i$ are all zeros and the diagonal entries are $\hat{z}_i - Z_i$, this establishes (3.9) with $\mu_i = 1$.

The other case is $|\hat{z}_i| > 2$. Take $\mu_i = \hat{z}_i/Z_i$, so $\hat{H}_i - \mu_i H_i$ has zeros in all diagonal entries, and the subdiagonal entries are all equal to $1 - \hat{z}_i/Z_i$, which simplifies to $\delta_i(1+\hat{z}_i)/(1+\delta_i)$. Thus, $\|\hat{H}_i - \mu_i H_i\| \leq 2\epsilon_{\text{mach}}(1+|\hat{z}_i|)/(1-2\epsilon_{\text{mach}})$, again establishing (3.9). We assume ϵ_{mach} is small enough so that $1/(1 - 2\epsilon_{\text{mach}}) < 1.5$.

Recall that $\hat{q}_B = \hat{H}_n \cdots \hat{H}_1$ while $Q_B = H_n \cdots H_1$. Let $\tau_2 = \mu_1 \cdots \mu_n$. Then

$$\begin{aligned}\|\hat{q}_B - \tau_2 Q_B\| &= \|\hat{H}_n \cdots \hat{H}_1 - (\mu_n H_n) \cdots (\mu_1 H_1)\| \\ &\leq \sum_{i=1}^{n} \|\hat{H}_n \cdots \hat{H}_i(\mu_{i-1} H_{i-1}) \cdots (\mu_1 H_1) - \hat{H}_n \cdots \hat{H}_{i+1}(\mu_i H_i) \cdots (\mu_1 H_1)\|\end{aligned}$$

$$\begin{aligned}
&\le \sum_{i=1}^{n} \|\hat{H}_n \cdots \hat{H}_{i+1}\| \cdot \|\hat{H}_i - \mu_i H_i\| \cdot \|(\mu_{i-1}H_{i-1}) \cdots (\mu_1 H_1)\| \\
&\le \sum_{i=1}^{n} (|\hat{z}_n|+1) \cdots (|\hat{z}_{i+1}|+1) \cdot 3\epsilon_{\text{mach}} \cdot (|\hat{z}_i|+1) \cdot (|\hat{z}_{i-1}|+1) \cdots (|\hat{z}_1|+1) \\
&= 3n\epsilon_{\text{mach}} (|\hat{z}_1|+1) \cdots (|\hat{z}_n|+1).
\end{aligned}$$

Here, the second line is obtained from the first via a telescoping-sum argument. The third is obtained from the second by factoring and using submultiplicativity. The fourth is obtained from the third using (3.7) for $\hat{H}_i$ and (3.9). To obtain the fourth line we also used $\|\mu_i H_i\| \le |\hat{z}_i|+1$. This follows from (3.7) because $\mu_i H_i = \hat{H}_i + O(\epsilon_{\text{mach}})$ and we are dropping second-order and higher terms.

On the other hand,

$$\begin{aligned}
\|\hat{q}_B\| &= \|\hat{H}_n \cdots \hat{H}_1\| \\
&\ge \sigma_{\min}(\hat{H}_n) \cdots \sigma_{\min}(\hat{H}_1) \\
&\ge \max(\sin(\pi/2), |\hat{z}_1|-1) \cdots \max(\sin(\pi/(n+1)), |\hat{z}_n|-1) \\
&\ge \max(1, |\hat{z}_1|-1) \cdots \max(2/(n+1), |\hat{z}_n|-1) \\
&\ge (|\hat{z}_1|+1)(1/3) \cdot (|\hat{z}_2|+1)(2/9) \cdots (|\hat{z}_n|+1)(2/(3(n+1))).
\end{aligned}$$

Here, the second line was derived from the first since $\|AB\| \ge \sigma_{\min}(A)\sigma_{\min}(B)$ for matrices with full column rank. The third was derived from the second using (3.8). The fourth was derived from the third since $\sin x \ge 2x/\pi$ for $x \in [0, \pi/2]$. The fifth was derived from the relation $\max(a, |x|-1) \ge (a/3)(|x|+1)$ for $a \in [0,1]$, which is proved by taking cases.

Combining the last two chains of inequalities shows that

$$\|\hat{q}_B - \tau_2 Q_B\| \le 3n\epsilon_{\text{mach}} \cdot 3 \cdot (9/2) \cdots (3(n+1)/2) \cdot \|\hat{q}_B\|. \tag{3.11}$$

We have $\|q_B - \tau_2 \hat{q}_B\| \le C_n \|q_B\| \epsilon_{\text{mach}}$ using the result of Van Dooren and Dewilde. Combining this inequality with (3.11) and dropping second-order terms yields

$$\|q_B - \tau_1\tau_2 Q_B\| \le C_n' \epsilon_{\text{mach}} \|q_B\|, \tag{3.12}$$

where $C_n' = C_n + (3n)3 \cdot (9/2) \cdots (3(n+1)/2)$. In other words, the pencil method followed by the FLT is a stable method for solving univariate Bézier polynomials.

In previous work [8], we proposed a different pencil for solving Bézier polynomials, which is as follows. It is straightforward to show that t is a root of p_B if and only if $\bar{A} - t\bar{B}$ is singular, where

$$\bar{A} - \lambda\bar{B} = \begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ -b_0 & -b_1 & \cdots & -b_{n-2} & -b_{n-1} \end{bmatrix}$$

$$-\lambda \begin{bmatrix} \frac{n}{1} & 1 & & & \\ & \frac{n-1}{2} & 1 & & \\ & & \ddots & \ddots & \\ & & & \frac{2}{n-1} & 1 \\ -b_0 & -b_1 & \cdots & -b_{n-2} & -b_{n-1} + \frac{b_n}{n} \end{bmatrix}. \tag{3.13}$$

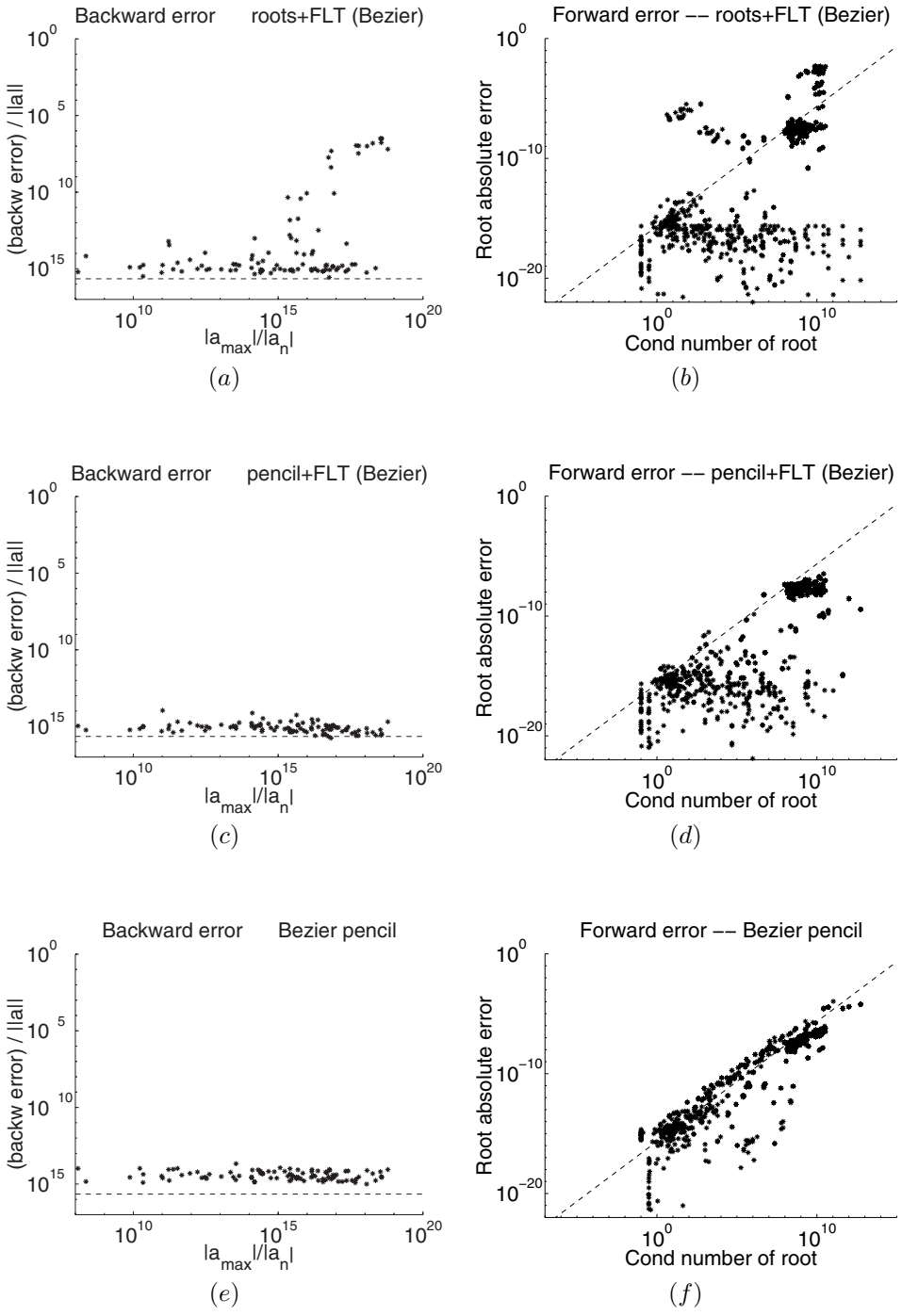


FIG. 3.2. *The backward and forward error of* roots *applied to q_B followed by the fractional linear transformation $t = z/(z+1)$ is depicted in* (*a*) *and* (*b*). *In* (*c*) *and* (*d*) *we show the algorithm combining the Van Dooren–Dewilde pencil* (1.3) *with FLT. Finally, in* (*e*) *and* (*f*) *we show the results for the pencil* (3.13) *that gives the Bézier roots directly without requiring an FLT.*

We found two ways to analyze this pencil algorithm (3.13): an analysis along the lines of Edelman and Murakami and another along the lines Van Dooren and Dewilde. Both analyses are written in [8]. The upshot is that we obtain a bound like (3.12) except the constant factor C_n' is better in the analysis for (3.13) as it is exponential rather than super-exponential. The algorithm based on (3.13) also has the advantage that the code is slightly simpler, since there is no need for an *if* statement to handle the case when a computed eigenvalue z is exactly equal to -1. The two algorithms are similar in terms of numerical results presented in the next section, so the difference in constant factors is more likely a shortcoming in the analysis in this section rather than in the algorithm itself.

3.1. Numerical results. Recall the formula (2.2) for the condition number of a root of a polynomial. We need to modify it to work for Bézier polynomials. If we consider $\mathbf{b} = [b_0, \ldots, b_n]$ as the input data, we get by a derivation similar to that of [13] that the condition number of the root τ of p_B is

$$\kappa_{\text{bez}}(\tau, p_B) = \|\mathbf{b}\| \frac{\|\tilde{\boldsymbol{\tau}}\|}{|p_B'(\tau)|}, \tag{3.14}$$

where $\tilde{\boldsymbol{\tau}} = [\binom{n}{0}(1-\tau)^n, \binom{n}{1}(1-\tau)^{n-1}\tau, \ldots, \binom{n}{n}\tau^n]$. Farouki and Rajan [4] defined a similar condition number for Bézier polynomial evaluation and root-finding, except their condition number is in terms of componentwise perturbation of coefficient rather than normwise. We return to the issue of componentwise versus normwise in section 4.

Figure 3.2 shows the forward error for the roots of 100 random Bézier polynomials of degree 10, where the coefficients b_j are chosen as follows. Instead of choosing the control points, we chose the coefficients of the matrix q_B described in the last section. First, we computed a polynomial q_5 of degree 5 with leading coefficient 10^{-10} and remaining coefficients of the form $(\alpha + i\beta) \cdot 10^{\gamma}$ with α, β chosen uniformly at random in $[-1, 1]$ and γ chosen uniformly in $[-10, 10]$. Then $q_B(z) = q_5(z)(z - 1/2)^2(z - .4)^2(z + 1)$. The factor $z + 1$ was included in the definition of q_B to show that the method in the last section still works well even if -1 is a root of q_B. The control points are then obtained from the coefficients of q_B by dividing by the binomial coefficients $\binom{n}{i}$.

We see that roots computed using either the Van Dooren–Dewilde pencil (1.3) followed by an FLT or the pencil (3.13) have forward errors around $\kappa_{\text{bez}}\epsilon_{\text{mach}}$ or smaller. But this does not hold for the roots computed using `roots`. So, just as for the standard polynomials, there are accuracy benefits by using the pencil.

4. Computing large roots and componentwise bounds. The focus so far has been on computing the smaller roots, which are the most important for geometric modeling. In this section we consider briefly the matter of computing large roots accurately. To focus the discussion in this section, consider the specific degree-10 polynomial $p(x)$ obtained by multiplying $10^{-10}x^8 + .2x^6 + .3^5 - .5x^4 + x^3 - 2.2x^2 - 1.2x - .3$ by $(x - .5)^2$. All the roots of this polynomial have absolute value less than 10 except for a pair of conjugate roots approximately $.75 \pm 4.47 \cdot 10^4 i$. The pencil algorithm for this polynomial gets the small roots with absolute and relative accuracy of about ϵ_{mach}, but the two larger roots are obtained with relative accuracy of only about 10^{-8}, i.e., absolute accuracy about 10^{-4}. The inaccurate computation of these larger roots does not violate our theory, since these roots are ill conditioned according to our definition (2.2). The large roots can be obtained to full machine precision by first substituting $x = 10^3 y$ into $p(x) = 0$, obtaining a new polynomial equation

$q(y) = 0$. Then the roots are computed for q using the pencil, and the resulting roots are all scaled by 10^3. This polynomial q no longer has a small leading coefficient. This transformation has the effect of making the large roots well conditioned, and they are now computed with absolute error smaller than 10^{-10}. On the other hand, this transformation has the side effect of making the smaller roots ill conditioned, so they are now computed with only a few digits of accuracy. It is interesting to ask whether there is a method that gets both the large and small roots of this polynomial accurately in a single computation using ordinary floating-point arithmetic.

We noticed in our test runs (not plotted) that `roots` often gets the large roots much more accurately than the pencil algorithm. The reason is that `roots` uses the `eig` function of Matlab, which in turn calls a *balancing* routine. Balancing means multiplying the companion matrix A on the left by D and on the right by D^{-1} for some diagonal matrix D chosen to make the entries of A better scaled [11]. One possible choice of D would be a diagonal matrix of the form $\mathrm{diag}(1, \alpha, \alpha^2, \ldots, \alpha^{n-1})$, which has the effect of replacing $p(x)$ by $p(x/\alpha)$ and thus carrying out the scaling described in the last paragraph. The actual balancing matrix is not of this form, but nonetheless the presence of balancing explains why it might be possible for `roots` to sometimes be especially accurate for large roots. On the other hand, balancing does not always work well—`roots` returned very inaccurate answers for the polynomial in the last paragraph, getting neither the larger nor smaller roots to more than four digits of accuracy.

There is some additional unexpected accuracy in the QR and QZ that we are currently not able to explain. In particular, both plots in Figure 2.2(a) and (b) show roots far below the dotted line. These are not large roots (since large roots were excluded from the plots). In addition, we carried out some additional test runs (not shown) involving eigenvalues without balancing, i.e., we used the `nobalance` option to the `eig` function. These smaller roots were still found with unexpected accuracy, indicating that balancing is not the whole story.

The QZ algorithm in Matlab used for pencils does not carry out any balancing. This is because balancing is not well understood for the QZ method, and there is no generally accepted method.

It is a bit unsettling that the condition number of the roots of a polynomial can change so drastically under a rescaling of the unknown like $x = y/1000$, which seemingly does not really change the underlying polynomial. An alternative not affected by scaling is to define a condition number for polynomials in terms of componentwise perturbations of the coefficients instead of normwise perturbations. In other words, the condition number of a root x_1 of $p(x) = a_n x^n + \cdots + a_0$ is defined to be the amount that the root changes when each a_i is perturbed to $a_i(1 + \delta_i)$ for some small choices of $\delta_0, \ldots, \delta_n$. Clearly this condition number is unchanged by a rescaling. Componentwise condition numbers were used by Toh and Trefethen and also by Farouki and Rajan. The ideal rootfinder would be one whose backward error bound is small in the componentwise sense, i.e., the roots computed by the algorithm are the exact roots to a polynomial close to the original polynomial in a componentwise sense. It is implausible that such a rootfinder could exist. For example, for the polynomial $x^n - 1$ it would have to return exact roots to $x^n - (1 + \epsilon)$ for a small ϵ, i.e., it would have return floating-point roots with exactly evenly spaced args and with exactly the same absolute values, which appears to be impossible. Perhaps more realistic would be to ask for an algorithm whose forward error is at most the componentwise condition number multiplied by ϵ_{mach}. We don't know of such an algorithm, but we don't know of a counterexample that would rule out its existence.

Acknowledgments. We thank the two anonymous referees for their helpful comments. In particular, the original version of the manuscript did not consider the algorithm analyzed in section 3 because we had presumed it would be unstable. One of the referee's remarks caused us to reconsider the matter.

REFERENCES

[1] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
[2] A. Edelman and H. Murakami, *Polynomial roots from companion matrix eigenvalues*, Math. Comp., 64 (1995), pp. 763–776.
[3] G. Farin, *Curves and Surfaces for Computer-Aided Geometric Design*, 4th ed., Academic Press, San Diego, 1997.
[4] R. T. Farouki and V. T. Rajan, *On the numerical condition of polynomials in Bernstein form*, Comput. Aided Geom. Design, 4 (1987), pp. 191–216.
[5] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., John Hopkins University Press, Baltimore, 1996.
[6] M. T. Heath, *Scientific Computing: An Introductory Survey*, McGraw-Hill, New York, 1997.
[7] N. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
[8] G. Jónsson, *Eigenvalue Methods for Accurate Solution of Polynomial Equations*, Ph.D. thesis, Center for Applied Mathematics, Cornell University, Ithaca, NY, 2001.
[9] S. A. Mitchell and S. A. Vavasis, *Quality mesh generation in higher dimensions*, SIAM J. Comput., 29 (2000), pp. 1334–1370.
[10] C. Moler, *Cleve's corner: ROOTS—of polynomials, that is*, The MathWorks Newsletter, 5 (1991), pp. 8–9.
[11] B. N. Parlett and C. Reinsch, *Balancing a matrix for calculation of eigenvalues and eigenvectors*, Numer. Math., 13 (1969), pp. 293–304.
[12] F. Tisseur, *Backward Stability of the QR Algorithm*, Tech. Report 239, UMR 5585 Lyon Saint-Etienne, October 1996.
[13] K.-C. Toh and L. N. Trefethen, *Pseudozeros of polynomials and pseudospectra of companion matrices*, Numer. Math., 68 (1994), pp. 403–425.
[14] P. Van Dooren and P. Dewilde, *The eigenstructure of an arbitrary polynomial matrix: Computational aspects*, Linear Algebra Appl., 50 (1983), pp. 545–579.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 2, pp. 415–425

A SIMPLE AND LESS CONSERVATIVE TEST FOR $\mathbb{D}$-STABILITY*

RICARDO C. L. F. OLIVEIRA† AND PEDRO L. D. PERES†

Abstract. This paper is concerned with the characterization of the Hurwitz stability of matrices, which can be written as the product of two square matrices AD with A precisely known and D belonging to the set of all positive diagonal matrices, and the Schur stability of matrices AD for all diagonal D, whose entries have absolute magnitude less than or equal to 1, known as the problem of $\mathbb{D}$-stability. Sufficient conditions are given in terms of linear matrix inequalities formulated at the vertices of an adequately chosen polytope domain, allowing simple and numerically efficient evaluations of $\mathbb{D}$-stability. The conditions proposed provide less conservative results and encompass previous conditions from the literature, as illustrated by examples.

Key words. Hurwitz stability, Schur stability, $\mathbb{D}$-stability, linear matrix inequalities

AMS subject classifications. 93D05, 93D09, 90C22, 15A39

DOI. 10.1137/S0895479803433842

Notation. P_D indicates that matrix P is diagonal. A diagonal matrix $n \times n$ is described as $\mathrm{diag}\{\sigma_{11}, \ldots, \sigma_{nn}\}$. The symbol $(')$ indicates transpose; $P > 0$ (≥ 0) means that P is symmetric positive (semi)definite and $P > M$ means that $P - M$ is symmetric positive definite. $\mathrm{Tr}(A)$ is the trace of the $n \times n$ matrix A, a_{ij} are its entries, and $\lambda_i(A)$, $i = 1, \ldots, n$, are its n eigenvalues.

1. Introduction.

The characterization of Hurwitz or Schur stability of matrices belonging to a set such as a polytope or to sets defined by interval matrices has been addressed in many papers. Although small dimension sets or special cases have been identified as Hurwitz (Schur) stable by means of algebraic methods or thanks to some particular property of the matrices, necessary and sufficient conditions which can be efficiently tested are still under investigation.

A set of particular importance is defined by the product of square matrices AD, with A precisely known and D belonging to the set of diagonal positive matrices. Matrices A such that all matrices AD with positive diagonal D are Hurwitz (i.e., all the eigenvalues of AD have negative real part) are called $\mathbb{D}$-stable matrices [22]. In the context of Schur stability, A is $\mathbb{D}$-stable if all the eigenvalues of AD have absolute magnitude less than 1 for all diagonal D such that $\mid d_{jj} \mid \leq 1$.

The importance of $\mathbb{D}$-stability can be inferred by the large number of papers dealing with this subject, in fields such as economics, biology, multiparameter singular perturbations, and large scale and decentralized control systems [23], [28], [31], [43]. As discussed in [26], the concept of $\mathbb{D}$-stability of a matrix was introduced in mathematical economics [1], [38], being used later in mathematical ecology. Other applications of $\mathbb{D}$-stability can be found in [3], [4], [9], [10], [14], [20], [21], [38], [41]. For instance, a generalized multispecies Lotka–Volterra model can be described by

$$\dot{x}_i(t) = c_i x_i(t) + \sum_{j=1}^{n} a_{ij} x_i(t) x_j(t)$$

*Received by the editors August 20, 2003; accepted for publication (in revised form) by A. H. Sayed March 1, 2004; published electronically January 12, 2005. This work was partially supported by the Brazilian agencies FAPESP, CAPES, and CNPq.
http://www.siam.org/journals/simax/26-2/43384.html
†School of Electrical and Computer Engineering, University of Campinas, CP 6101, 13081-970, Campinas SP, Brazil (ricfow@dt.fee.unicamp.br, peres@dt.fee.unicamp.br).

for $i = 1, \ldots, n$. Similar models are used in problems like the management of fish populations, the spread of epidemics, the propagation of genetic traits, and the kinetics of autocatalytic chemical reactions. For details, references, and other applications, the reader is referred to [28]. Assuming that $x_i(t) \neq 0$, $i = 1, \ldots, n$ (that is, no species disappears), one has

$$\dot{x}_i(t)/x_i(t) = c_i + \sum_{j=1}^{n} a_{ij} x_j(t)$$

or, equivalently,

$$\frac{d}{dt}[\ln x_i(t)] = c_i + \sum_{j=1}^{n} a_{ij} \exp[\ln x_j(t)].$$

Defining $y_i(t) \triangleq \ln x_i(t)$, one has

$$\dot{y}_i(t) = c_i + \sum_{j=1}^{n} a_{ij} f(y_j(t)).$$

Assuming the existence of a positive equilibrium y^*, i.e., $y^* = \begin{bmatrix} y_1^* & \cdots & y_n^* \end{bmatrix}'$, with $y_i > 0$ for all i, such that $c_i + \sum_{j=1}^{n} a_{ij} f(y_j^*) = 0$, one has the incremental model

$$\dot{z}_i(t) = \sum_{j=1}^{n} a_{ij} g_j(z_j(t)),$$

where, for all i, $z_i(t) \triangleq y_i(t) - y_i^*$; $z(t) = \begin{bmatrix} z_1(t) & \cdots & z_n(t) \end{bmatrix}'$, $g_i : \xi \to \exp(\xi + \ln x_i^*) - x_i^*$; therefore, $g_i(0) = 0$ and $g_i(\xi)\xi > 0$ for $\xi \neq 0$. In this case, the $\mathbb{D}$-stability of the matrix $A \triangleq [a_{ij}]$ of the coefficients of interaction implies that the stability of the positive equilibrium is preserved under any variation of the equilibrium values of population sizes [26].

Since the concept of $\mathbb{D}$-stability has been introduced, much effort has been made to characterize classes of matrices that are $\mathbb{D}$-stable and also to provide less conservative sufficient conditions assuring Hurwitz or Schur $\mathbb{D}$-stability [6], [11], [15], [24], [33].

In fact, $\mathbb{D}$-stability can be classified as a particular case of the problem of global stability for linear systems with dynamics represented by matrix A subjected to state perturbations of multiplicative type [30]. A well-known sufficient condition for $\mathbb{D}$-stability is the existence of a positive definite diagonal Lyapunov matrix P_D such that $A'P_D + P_D A < 0$ (or $A'P_D A - P_D < 0$ in the discrete-time case), which is in fact a strong condition assuring global stability of the origin of a dynamic system represented by matrix A for a class of state perturbations (known as stability of Persidskii [37] and its corresponding extension to the discrete-time case [29]). Although there exist some classes of matrices, usually of small dimensions, for which the necessary and sufficient conditions for $\mathbb{D}$-stability are well established (for instance, there are constructive necessary and sufficient conditions of $\mathbb{D}$-stability for matrices of the second and third orders) [23], [27], the complete characterization of $\mathbb{D}$-stable matrices remains an open problem [22]. The difficulty lies mainly on the fact that algebraic methods become very involved as the order of matrix A increases [26], while the existing numerical algorithms, although more suitable for handling higher order systems, provide only sufficient conditions for $\mathbb{D}$-stability.

In the last two decades, a lot of work has been done to characterize robust stability for linear uncertain systems, and one of the major approaches is the simultaneous stability by means of a common Lyapunov function (also known as quadratic stability) [2]. From this condition, many robust control and filtering problems have been addressed by means of linear matrix inequalities (LMIs) [5], [7], [17], [19], [25], [32], [34], [35], which nowadays can be efficiently solved by polynomial time algorithms [16]. It is worth mentioning the recent works dealing with sum-of-squares representation [36], relaxations [42], and homogeneous Lyapunov functions [12], which investigate sharper characterizations of robust stability by means of LMIs.

The robust stability of compact sets such as polytopes can also be used to assess $\mathbb{D}$-stability, as proposed in [27] for discrete-time systems. In this case, a polytope AD is constructed, with D belonging to an adequately defined compact set, and the robust stability of the polytope is a necessary and sufficient condition for the Schur $\mathbb{D}$-stability of A. Similar results appeared in [18], [28] for the Hurwitz $\mathbb{D}$-stability, but the closed polytope tested was only an approximation of the open set of interest.

As a consequence, the existence of a common Lyapunov matrix assessing the robust stability of AD becomes another sufficient condition for $\mathbb{D}$-stability, formulated as a set of LMIs described at a finite number of vertices of an appropriate polytopic set [8], [27], [28]. However, the results can be very conservative, since the Lyapunov matrix is fixed. New conditions for structural and robust stability formulated as LMIs appeared in [13], [18]. The main idea was the introduction of some extra variables in the LMIs, allowing the characterization of $\mathbb{D}$-stability by means of a parameter-dependent Lyapunov function, which provides less conservative results than a fixed one.

In this paper, new simple and efficient LMI conditions for $\mathbb{D}$-stability are given. Using the ideas from [6], [13], [18], [27], combined with recent robust stability tests [39], [40], a parameter-dependent Lyapunov function is constructed and used to assess the Hurwitz (Schur) stability of a conveniently constructed polytope. These new conditions identify many $\mathbb{D}$-stable matrices that are not identified by other methods and also contain the previous conditions (quadratic and LMIs from [13], [18]) as particular cases. Both Hurwitz and Schur $\mathbb{D}$-stability are investigated and illustrated by examples.

Although the focus of this paper is on the stability of real matrices, the lemmas and theorems presented could be extended to deal with $\mathbb{D}$-stability of complex matrices, provided that Hermitian matrices P_i, $i = 1, \ldots, N$, and complex matrices G_i, H_i, $i = 1, \ldots, N$, were investigated as feasible solution of the complex LMIs.

2. Hurwitz $\mathbb{D}$-stability. In the context of Hurwitz stability, $\mathbb{D}$-stability can be viewed as a particular case of the more general problem of stability of dynamical systems described by the differential equation $\dot{x} = Af(x)$. It has been shown in [37] that the existence of a positive diagonal matrix P_D such that $A'P_D + P_DA < 0$ assures the global asymptotic stability of the equilibrium point $x = 0$ for a class of functions $f(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ belonging to a given set (see also [18], [27]). The existence of such a diagonal matrix P_D, sometimes called absolute stability, assures the robust stability for any function $f(x)$ such that $f_j(x_j)x_j > 0$, $j = 1, \ldots, n$ (see [28] for details). This includes functions of the form $f(x) = Dx$, where $D \in \mathbb{R}^{n\times n}$ is any positive diagonal matrix. Therefore, the existence of diagonal P_D implies that all matrices AD with positive diagonal D are Hurwitz (but the converse is not necessarily true). Note that this sufficient condition can be easily verified by testing the feasibility of the LMI $A'P_D + P_DA < 0$, which can be efficiently performed nowadays (easily handling the constraint $P = P_D$) by any available LMI solver [16], [44].

In order to obtain a sharper characterization of $\mathbb{D}$-stability, some properties of sets of Hurwitz matrices can be exploited. If AD is Hurwitz stable for any positive diagonal matrix D, then for any scalar $\rho > 0$ one has $\lambda_i(\rho AD) = \rho\lambda_i(AD)$, $i = 1, \ldots, n$, implying that ρAD also defines a set of Hurwitz stable matrices for any positive diagonal matrix D and $\rho > 0$. With that on mind, the test of Hurwitz stability of AD can be constrained to the set of positive diagonal matrices D such that $\mathrm{Tr}(D) = 1$, for instance.

This idea has been used in [18] to conveniently construct a convex polytope $\mathcal{D}_H$ by defining its vertex matrices as D_i, $i = 1, \ldots, N$, as positive diagonal matrices with unitary trace, whose entries are parametrized by a sufficiently small scalar $\epsilon > 0$. Then the Hurwitz stability of the set $\mathcal{D}_H$ (i.e., the Hurwitz stability of any matrix belonging to the set) is investigated by means of Lyapunov inequalities. As $\epsilon \to 0$, the Hurwitz stability of $\mathcal{D}_H$ tends to the $\mathbb{D}$-stability of A.

The problem of Hurwitz $\mathbb{D}$-stability, i.e., to determine if a given matrix $A \in \mathbb{R}^{n\times n}$ is such that AD is Hurwitz for all matrices positive and diagonal $D \in \mathbb{R}^{n\times n}$, can be addressed following the lines given in [18], [28], that is, by testing if the set of matrices AD is Hurwitz for all D belonging to the polytope $\mathcal{D}_H$ given by

$$\mathcal{D}_H = \left\{ D(\alpha) \ : \ D = \sum_{i=1}^{N} \alpha_i D_i \ ; \ \alpha_i \geq 0 \ ; \ \sum_{i=1}^{N} \alpha_i = 1 \right\}, \tag{2.1}$$

$$D_i = \mathrm{diag} \left\{ \frac{\epsilon}{n-1}, \ldots, \underbrace{1-\epsilon}_{i}, \ldots, \frac{\epsilon}{n-1} \right\} > 0, \quad i = 1, \ldots, N, \quad N = n, \tag{2.2}$$

with $\epsilon > 0$ arbitrarily small. As $\epsilon \to 0$, the Hurwitz stability of matrices AD, $D \in \mathcal{D}_H$ defined as the convex hull of matrices D_i given by (2.2), tends to the $\mathbb{D}$-stability of A (see [18] for details). In other words, the closed set defined in terms of ϵ approximates the open set of all positive diagonal matrices as $\epsilon \to 0$. Note that the number of vertices of this polytope equals the dimension of matrix A, and that any $D \in \mathcal{D}_H$ can be written as a convex combination of the vertex matrices D_i, being such that $\mathrm{Tr}(D) = 1$.

As discussed in [18], as $\epsilon \to 0$, the $\mathbb{D}$-stability of a matrix A can be inferred from the Hurwitz stability of the matrices AD such that $D \in \mathcal{D}_H$ given by (2.1)–(2.2), which is equivalent to the existence of a parameter-dependent positive definite Lyapunov matrix $P(\alpha)$ such that

$$(AD(\alpha))' P(\alpha) + P(\alpha) (AD(\alpha)) < 0 \tag{2.3}$$

holds for all AD with $D \in \mathcal{D}_H$.

A well-known sufficient condition for that comes from quadratic stability [2], that is, the same $P(\alpha) = P = P' > 0$ must verify

$$(AD_i)'P + P(AD_i) < 0, \quad i = 1, \ldots, N, \tag{2.4}$$

implying that (2.3) holds for all $D \in \mathcal{D}_H$. However, quadratic stability can be very conservative for testing the robust stability of the polytope defined by AD, $D \in \mathcal{D}_H$.

In [18], a less conservative evaluation of the Hurwitz stability is provided by means of a parameter-dependent Lyapunov matrix, which is reproduced in the next lemma.

LEMMA 2.1. *If there exist symmetric positive definite matrices $P_i \in \mathbb{R}^{n\times n}$, $i = 1,\dots,N$, matrices $G \in \mathbb{R}^{n\times n}$ and $H \in \mathbb{R}^{n\times n}$ satisfying the LMIs*

$$\begin{bmatrix} GD_i + D_i'G' & P_iA - G + D_i'H' \\ A'P_i - G' + HD_i & -H - H' \end{bmatrix} < 0, \quad i = 1,\dots,N, \tag{2.5}$$

then $P(\alpha) > 0$ given by

$$P(\alpha) = \sum_{i=1}^{N} \alpha_i P_i, \quad \alpha_i \geq 0, \quad \sum_{i=1}^{N} \alpha_i = 1 \tag{2.6}$$

is such that (2.3) *holds, implying that AD is Hurwitz for all $D \in \mathcal{D}_H$.*

Proof. See [18] for the proof. □

Note that if a feasible solution exists for (2.5), then a parameter-dependent Lyapunov function $v(x) = x'P(\alpha)x$ with $P(\alpha)$ given by (2.6) assures the Hurwitz stability of AD for all $D \in \mathcal{D}_H$. Moreover, the results of Lemma 2.1 encompass the fixed Lyapunov matrix (i.e., $P_i = P$, $i = 1,\dots,N$) as a particular case.

In what follows, a new LMI condition assuring that (2.3) holds with $P(\alpha)$ given by (2.6) is presented. Besides being more general than the results of Lemma 2.1, containing (2.5) as a particular case, the conditions provide less conservative evaluations of Hurwitz stability of AD, $D \in \mathcal{D}_H$, and, consequently, of $\mathbb{D}$-stability.

THEOREM 2.2. *If there exist symmetric positive definite matrices $P_i \in \mathbb{R}^{n\times n}$, matrices $G_i \in \mathbb{R}^{n\times n}$ and $H_i \in \mathbb{R}^{n\times n}$, $i = 1,\dots,N$, satisfying the LMIs*

$$\begin{bmatrix} G_iD_i + D_i'G_i' & P_iA - G_i + D_i'H_i' \\ A'P_i - G_i' + H_iD_i & -H_i - H_i' \end{bmatrix} < -\mathbf{I}, \quad i = 1,\dots,N, \tag{2.7}$$

$$\begin{bmatrix} G_iD_j + D_j'G_i' + G_jD_i + D_i'G_j' \\ A'(P_i + P_j) - G_i' - G_j' + H_iD_j + H_jD_i \\ (P_i + P_j)A - G_i - G_j + D_i'H_j' + D_j'H_i' \\ -H_i - H_i' - H_j - H_j' \end{bmatrix} < \frac{2}{N-1}\mathbf{I}, \tag{2.8}$$

$$i = 1,\dots,N-1, \quad j = i+1,\dots,N,$$

then (2.3) *holds with $P(\alpha) > 0$ given by* (2.6)*, implying that AD is Hurwitz for all $D \in \mathcal{D}_H$.*

Proof. Multiplying (2.7) by α_i^2 and summing for $i = 1,\dots,N$; multiplying (2.8) by $\alpha_i\alpha_j \geq 0$ and summing for $i = 1,\dots,N-1$, $j = i+1,\dots,N$; and taking into account (2.6) and defining $G(\alpha)$ and $H(\alpha)$,

$$G(\alpha) = \sum_{i=1}^{N} \alpha_i G_i, \quad H(\alpha) = \sum_{i=1}^{N} \alpha_i H_i, \quad \alpha_i \geq 0, \quad \sum_{i=1}^{N} \alpha_i = 1, \tag{2.9}$$

one has (see [40])

$$
\begin{aligned}
(2.10)\quad & \sum_{i=1}^{N} \alpha_i^2 \begin{bmatrix} G_i D_i + D_i' G_i' & P_i A - G_i + D_i' H_i' \\ A' P_i - G_i' + H_i D_i & -H_i - H_i' \end{bmatrix} \\
& + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \alpha_i \alpha_j \begin{bmatrix} G_i D_j + D_j' G_i' + G_j D_i + D_i' G_j' \\ A'(P_i + P_j) - G_i' - G_j' + H_i D_j + H_j D_i \end{bmatrix} \\
& \qquad \begin{matrix} (P_i + P_j) A - G_i - G_j + D_i' H_j' + D_j' H_i' \\ -H_i - H_i' - H_j - H_j' \end{matrix} \Bigg] \\
& = \Theta_H(\alpha) \triangleq \begin{bmatrix} G(\alpha) D(\alpha) + D(\alpha)' G(\alpha)' & P(\alpha) A - G(\alpha) + D(\alpha)' H(\alpha)' \\ A' P(\alpha) - G(\alpha)' + H(\alpha) D(\alpha) & -H(\alpha) - H(\alpha)' \end{bmatrix} \\
& \qquad < - \left(\sum_{i=1}^{N} \alpha_i^2 - \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \alpha_i \alpha_j \frac{2}{N-1} \right) \mathbf{I} \leq 0,
\end{aligned}
$$

implying that $\Theta_H(\alpha) < 0$. Now, multiply $\Theta_H(\alpha)$ by $T(\alpha) = \begin{bmatrix} \mathbf{I} & D(\alpha)' \end{bmatrix}$ on the left and by $T(\alpha)'$ on the right to obtain (2.3), which guarantees that AD is Hurwitz for all $D \in \mathcal{D}_H$, since $P(\alpha)$ given by (2.6) with $P_i > 0$ is a parameter-dependent positive definite matrix. □

Theorem 2.2 provides sufficient conditions to test whether A is $\mathbb{D}$-stable by testing whether AD is Hurwitz for all $D \in \mathcal{D}_H$. As discussed in [18], as $\epsilon \to 0$ in (2.2), the Hurwitz stability of AD tends to the $\mathbb{D}$-stability of A. The feasibility of the LMIs (2.7)–(2.8) can be verified by polynomial time algorithms as, for instance, the interior point method in [16]. Besides being less conservative than the results presented in Lemma 2.1, the condition (2.7) reduces to (2.5) when $G_i = G$, $H_i = H$, $i = 1, \ldots, N$ (in this case, (2.8) holds whenever (2.7) is verified). Note that the right-hand side of the LMIs (2.7) can be arbitrarily fixed as $-\mathbf{I}$ thanks to the property of homogeneity, and that the identity matrix in (2.7)–(2.8) could be replaced by any other positive definite matrix with appropriate dimensions, yielding equivalent conditions provided that the coefficients are chosen in such a way that (2.10) holds. Moreover, as illustrated by means of examples, the results of Theorem 2.2 are able to identify Hurwitz $\mathbb{D}$-stable matrices for which the conditions of Lemma 2.1 do not hold.

3. Schur $\mathbb{D}$-stability. The problem of Schur $\mathbb{D}$-stability—i.e., to determine if a given matrix $A \in \mathbb{R}^{n \times n}$ is such that AD is Schur stable for all diagonal matrices D whose entries d_{jj} are such that $\mid d_{jj} \mid \leq 1$, $j = 1, \ldots, n$—can be addressed following the lines given in [6], [13], that is, by testing if matrices AD are Schur stable for all $D \in \mathcal{D}_S$ given by

$$
(3.1)\qquad \mathcal{D}_S = \left\{ D(\alpha) \ : \ D = \sum_{i=1}^{N} \alpha_i D_i \ ; \ \alpha_i \geq 0 \ ; \ \sum_{i=1}^{N} \alpha_i = 1 \right\},
$$

$$
(3.2)\qquad D_i = \text{diag}\,\{\sigma_1, \ldots, \sigma_j, \ldots, \sigma_n\}, \quad i = 1, \ldots, N, \quad N = 2^{n-1},
$$

with $\sigma_j \in \{-1, 1\}$, $j = 1, \ldots, n$, and $D_k \neq -D_\ell$ for all $k, \ell \in \{1, \ldots, 2^{n-1}\}$. Note that, similarly to the set $\mathcal{D}_H$ defined in the Hurwitz case, any $D \in \mathcal{D}_S$ can be written as a convex combination of the vertices D_i given by (3.2).

As discussed in [13], this choice of vertices generates a polytope which entirely represents the set of diagonal matrices whose entries have absolute value less than or

equal to 1 for Schur stability purposes, since for any scalar $\rho > 0$ one has $\rho\lambda_j(A) = \lambda_j(\rho A)$, $j = 1, \ldots, n$, implying that if $\mid \lambda_j(AD) \mid < 1$, $j = 1, \ldots, n$, for all $D \in \mathcal{D}_S$ given by (3.1)–(3.2), then $\mid \lambda_j(AD) \mid < 1$, $j = 1, \ldots, n$, for any diagonal matrix D whose entries are such that $\mid d_{ij} \mid \leq 1$. Note that the number of vertices of $\mathcal{D}_S$ is 2^{n-1}, while in $\mathcal{D}_H$ (i.e., the Hurwitz stability case) it was n, illustrating how the Schur $\mathbb{D}$-stability characterization is more involved.

The aim is to characterize the Schur stability of matrices AD for $D \in \mathcal{D}_S$ given by (3.1)–(3.2), that is, to determine a parameter-dependent Lyapunov matrix $P(\alpha) > 0$ such that

$$(AD(\alpha))' P(\alpha) (AD(\alpha)) - P(\alpha) < 0 \tag{3.3}$$

holds for all AD with $D \in \mathcal{D}_S$. By Schur complement, (3.3) is equivalent to

$$\begin{bmatrix} -P(\alpha) & P(\alpha)AD(\alpha) \\ D(\alpha)'A'P(\alpha) & -P(\alpha) \end{bmatrix} < 0. \tag{3.4}$$

The simplest way to deal with condition (3.3) is to impose that the same $P(\alpha) = P > 0$ must verify

$$(AD_i)'P(AD_i) - P < 0, \quad i = 1, \ldots, N, \tag{3.5}$$

or, equivalently,

$$\begin{bmatrix} -P & PAD_i \\ D_i'A'P & -P \end{bmatrix} < 0, \quad i = 1, \ldots, N, \tag{3.6}$$

implying that (3.3) holds for all AD with $D(\alpha) \in \mathcal{D}_S$. As in the Hurwitz case, this choice can produce conservative results in the evaluation of the Schur stability of AD since the same P is imposed for the entire set $\mathcal{D}_S$. A parameter-dependent Lyapunov function was proposed in [13] as an alternative way to investigate the $\mathbb{D}$-stability of matrix A, introducing some extra variables in the characterization of the Schur stability of the polytope defined by matrices AD, $D \in \mathcal{D}_S$.

LEMMA 3.1. *If there exist symmetric positive definite matrices $P_i \in \mathbb{R}^{n\times n}$, $i = 1, \ldots, N$, matrices $G \in \mathbb{R}^{n\times n}$ and $H \in \mathbb{R}^{n\times n}$ satisfying the LMIs*

$$\begin{bmatrix} GD_i + D_i'G' - P_i & D_i'H' - G \\ HD_i - G' & A'P_iA - H - H' \end{bmatrix} < 0, \quad i = 1, \ldots, N, \tag{3.7}$$

then $P(\alpha) > 0$ given by (2.6) *is such that* (3.3) *holds, implying that AD is Schur stable for all $D \in \mathcal{D}_S$.*

Proof. See [13] for the proof. □

With the results of Lemma 3.1, a parameter-dependent Lyapunov function given by $v(x(k)) = x(k)'P(\alpha)x(k)$ with $P(\alpha)$ as in (2.6) can be used to assess the Schur stability of AD for all $D \in \mathcal{D}_S$. Besides being less conservative than the results obtained through a fixed Lyapunov matrix (which can be recovered by fixing $P_i = P$, $i = 1, \ldots, N$), the fact that the same matrices G and H must satisfy all the LMIs makes the conditions of Lemma 3.1 far from the necessity. A more general condition is presented in what follows.

THEOREM 3.2. *If there exist symmetric positive definite matrices $P_i \in \mathbb{R}^{n\times n}$, matrices $G_i \in \mathbb{R}^{n\times n}$ and $H_i \in \mathbb{R}^{n\times n}$, $i = 1, \ldots, N$, satisfying the LMIs*

$$\begin{bmatrix} G_iD_i + D_i'G_i' - P_i & D_i'H_i' - G_i \\ H_iD_i - G_i' & A'P_iA - H_i - H_i' \end{bmatrix} < -\mathbf{I}, \quad i = 1, \ldots, N, \tag{3.8}$$

$$
\begin{bmatrix} G_iD_i + D_i'G_i' + G_iD_j + D_j'G_i' + G_jD_i + D_i'G_j' - 2P_i - P_j & D_i'(H_i' + H_j') + D_j'H_i' - 2G_i - G_j \\ (H_i + H_j)D_i + H_iD_j - 2G_i' - G_j' & 2A'P_iA + A'P_jA - 2H_i - 2H_i' - H_j - H_j' \end{bmatrix} < \frac{1}{(N-1)^2}\mathbf{I}, \tag{3.9}
$$

$$
i = 1, \ldots, N, \quad j \neq i, \quad j = 1, \ldots, N,
$$

$$
\begin{bmatrix} G_iD_j + D_j'G_i' + G_iD_k + D_k'G_i' + G_jD_i + D_i'G_j' + G_jD_k + D_k'G_j' \\ +G_kD_i + D_i'G_k' + G_kD_j + D_j'G_k' - 2(P_i + P_j + P_k) & \\ & \\ H_jD_i + H_kD_i + H_iD_j + H_kD_j & D_i'H_j' + D_i'H_k' + D_j'H_i' + D_j'H_k' \\ +H_iD_k + H_jD_k - 2(G_i + G_j + G_k)' & +D_k'H_i' + D_k'H_j' - 2(G_i + G_j + G_k) \\ & \\ 2(A'P_iA + A'P_jA + A'P_kA) & \\ -2(H_i + H_i' + H_j + H_j' + H_k + H_k') & \end{bmatrix} \tag{3.10}
$$

$$
< \frac{6}{(N-1)^2}\mathbf{I}, \qquad \begin{matrix} i = 1, \ldots, N-2, \\ j = i+1, \ldots, N-1, \quad k = j+1, \ldots, N, \end{matrix}
$$

then (3.3) *holds with* $P(\alpha) > 0$ *given by* (2.6)*, implying that* AD *is Schur stable for all* $D \in \mathcal{D}_S$*.*

Proof. The proof follows similar steps to those of Theorem 2.2. Multiply (3.8) by α_i^3 and sum for $i = 1, \ldots, N$; multiply (3.9) by $\alpha_i^2\alpha_j$ and sum for $i = 1, \ldots, N$; and multiply $j \neq i, j = 1, \ldots, N$ and (3.10) by $\alpha_i\alpha_j\alpha_k$, $i = 1, \ldots, N-2$, $j = i+1, \ldots, N-1$, $k = j+1, \ldots, N$, with $G(\alpha)$ and $H(\alpha)$ as in (2.9), to obtain (see [39])

$$
\Theta_S(\alpha) \triangleq \begin{bmatrix} G(\alpha)D(\alpha) + D(\alpha)'G(\alpha)' - P(\alpha) & D(\alpha)'H(\alpha)' - G(\alpha) \\ H(\alpha)D(\alpha) - G(\alpha)' & A'P(\alpha)A - H(\alpha) - H(\alpha)' \end{bmatrix} \tag{3.11}
$$

$$
< -\left(\sum_{i=1}^{N} \alpha_i^3 - \frac{1}{(N-1)^2} \sum_{i=1}^{N} \sum_{j \neq i; j=1}^{N} \alpha_i^2\alpha_j - \frac{6}{(N-1)^2} \sum_{i=1}^{N-2} \sum_{j=i+1}^{N-1} \sum_{k=j+1}^{N} \alpha_i\alpha_j\alpha_k \right) \mathbf{I}.
$$

By premultiplying $\Theta_S(\alpha) < 0$ by $T(\alpha) = \begin{bmatrix} \mathbf{I} & D(\alpha)' \end{bmatrix}$ and by postmultiplying by $T(\alpha)'$, one gets (3.3), which guarantees that A is $\mathbb{D}$-stable. ∎

As in the Hurwitz case (Theorem 2.2), the LMI conditions of Theorem 3.2 can be solved by polynomial time algorithms. These conditions are less conservative than the ones presented in Lemma 3.1. In fact, condition (3.8) contains (3.7) as a particular case when $G = G_i$, $H = H_i$, $i = 1, \ldots, N$. In this case (i.e., $G = G_i$ and $H = H_i$) the LMIs (3.9)–(3.10) are always verified if (3.8) holds. Note that the right-hand side of the LMI (3.8) can be arbitrarily fixed as $-\mathbf{I}$ thanks to the property of homogeneity, which also assures that matrix $\mathbf{I}$ appearing at the right-hand side of the LMIs (3.8)–(3.10) could be replaced by any other positive definite matrix, yielding equivalent results provided the coefficients are such that (3.11) holds.

Finally, a remark about the larger number of LMIs obtained in the conditions of Theorem 3.2, when compared to the continuous-time case (Theorem 2.2). This is due to the lines followed in the paper (similar to the ones in [39]) to assure that (3.3) holds. Note that α_i^3 terms naturally appear in (3.3) for $D(\alpha)$ given by (3.1)–(3.2) and $P(\alpha)$ given by (2.6), and conditions (3.8)–(3.10) have been constructed in order to guarantee that (3.11) holds, that is, $\Theta_D(\alpha) < 0$, implying that (3.3) is satisfied.

4. Examples. The numerical tests were performed using the LMI Control Toolbox [16]. By means of randomly generated matrices, it is shown that the conditions proposed in the paper are able to identify $\mathbb{D}$-stability in cases where the other methods fail.

The first examples are concerned with Hurwitz $\mathbb{D}$-stability, and $\epsilon = 0.001$ has been used to construct the polytope $\mathcal{D}_H$ in (2.1)–(2.2). Although there exist algebraic characterizations for second and third order systems, the conditions of Lemma 2.1 and Theorem 2.2 have been tested for several small-size systems ($n = 2, 3$). For $n = 2$ all the examples of $\mathbb{D}$-stable matrices have been identified by both Lemma 2.1 and Theorem 2.2, but for $n \geq 3$ it is very simple to find examples of $\mathbb{D}$-stable matrices which are identified by Theorem 2.2 but not by Lemma 2.1 conditions, as in the case with the following matrices (which are Hurwitz stable but do not admit a diagonal positive solution P_D to the Lyapunov inequality $A'P_D + P_D A < 0$):

$$A = \begin{bmatrix} -3 & 4 & -2 \\ -1 & 0 & 0 \\ -1 & 3 & -4 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 0 & -1 \\ 1 & -2 & 0 \\ 2 & -1 & -2 \end{bmatrix},$$

$$A = \begin{bmatrix} 0 & -1 & 0 \\ 2 & -3 & -1 \\ 3 & -1 & -3 \end{bmatrix}, \quad A = \begin{bmatrix} -3 & -1 & 1 \\ 1 & -5 & 0 \\ -3 & 3 & 0 \end{bmatrix}.$$

These small-dimension examples are presented here only to illustrate that even in the cases where algebraic characterization exists, the conditions of Lemma 2.1 fail to guarantee $\mathbb{D}$-stability (but not Theorem 2.2). Moreover, Theorem 2.2 can easily evaluate the $\mathbb{D}$-stability of matrices with greater dimension, for which there is no other characterization available, as in the following matrices (which do not admit a diagonal positive solution P_D to the Lyapunov inequality $A'P_D + P_D A < 0$), identified as being $\mathbb{D}$-stable:

$$A = \begin{bmatrix} -7 & -2 & 4 & 1 \\ -1 & -4 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ -4 & -3 & 4 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} -4 & -1 & 2 & -1 \\ 0 & -4 & 3 & -2 \\ 0 & -1 & 0 & 0 \\ 0 & -1 & 1 & -3 \end{bmatrix},$$

$$A = \begin{bmatrix} -2 & 2 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ -1 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 1 \\ 0 & -1 & 0 & 0 & -2 \end{bmatrix}.$$

Now, Schur $\mathbb{D}$-stability is investigated. As discussed in [6], for any 2×2 matrix, the existence of a positive diagonal solution to the Lyapunov equation is a necessary and sufficient condition for Schur $\mathbb{D}$-stability, indicating that a larger dimension is needed to find a $\mathbb{D}$-stable matrix which is not diagonally stable. Moreover, in [15] it has been shown that for 3×3 matrices the vertex stability of the polytope defined by AD with $D \in \mathcal{D}_S$ is equivalent to the $\mathbb{D}$-stability of A.

For higher dimensions, there is not a characterization based on the vertices of AD, $D \in \mathcal{D}_S$, but the results of Theorem 3.2 can be efficiently used. For instance, the

following matrix A is Schur stable, but does not admit a diagonal positive solution P_D to the Lyapunov inequality $A'P_DA - P_D < 0$:

$$A = \begin{bmatrix} 0.13 & 0.26 & -0.54 & 0.12 \\ -0.63 & 0.59 & 0.29 & -0.13 \\ 0.22 & 0.26 & -0.68 & 0.18 \\ -0.78 & 1.10 & -0.04 & -0.19 \end{bmatrix}. \tag{4.1}$$

By constructing the polytope $\mathcal{D}_S$ as in (3.1)–(3.2), one may check that no feasible solution is obtained through the conditions of Lemma 3.1, but the conditions of Theorem 3.2 provide a positive evaluation of the Schur stability of the polytope, implying that A is Schur $\mathbb{D}$-stable.

5. Conclusion. Improved sufficient conditions for Hurwitz and Schur $\mathbb{D}$-stability have been given. The conditions are formulated as LMIs and can be efficiently solved by means of polynomial time algorithms, providing less conservative evaluations of $\mathbb{D}$-stability and encompassing previous results from the literature as particular cases.

Acknowledgment. The authors wish to thank the reviewers for their valuable suggestions.

REFERENCES

[1] K. J. Arrow and M. McManus, *A note on dynamic stability*, Econometrica, 26 (1958), pp. 448–454.
[2] B. R. Barmish, *Necessary and sufficient conditions for quadratic stabilizability of an uncertain system*, J. Optim. Theory Appl., 46 (1985), pp. 399–408.
[3] S. Barnett and C. Storey, *Some applications of the Lyapunov matrix equation*, J. Inst. Math. Appl., 4 (1968), pp. 33–42.
[4] A. Berman and D. Hershkowitz, *Matrix diagonal stability and its implications*, SIAM J. Algebraic Discrete Methods, 4 (1983), pp. 377–382.
[5] J. Bernussou, P. L. D. Peres, and J. C. Geromel, *A linear programming oriented procedure for quadratic stabilization of uncertain systems*, Systems Control Lett., 13 (1989), pp. 65–72.
[6] A. Bhaya and E. Kaszkurewicz, *On discrete-time diagonal and D-stability*, Linear Algebra Appl., 187 (1993), pp. 87–104.
[7] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM Stud. Appl. Math. 15, SIAM, Philadelphia, 1994.
[8] S. Boyd and Q. Yang, *Structured and simultaneous Lyapunov functions for system stability problems*, Internat. J. Control, 49 (1989), pp. 2215–2240.
[9] B. Cain, L. M. DeAlba, L. Hogben, and C. R. Johnson, *Multiplicative perturbations of stable and convergent operators*, Linear Algebra Appl., 268 (1998), pp. 151–169.
[10] B. E. Cain, *Inside the D-stable matrices*, Linear Algebra Appl., 56 (1984), pp. 237–243.
[11] J. Chen, M. K. H. Fan, and C. C. Yu, *On D-stability and structured singular values*, Systems Control Lett., 24 (1995), pp. 19–24.
[12] G. Chesi, A. Garulli, A. Tesi, and A. Vicino, *Homogeneous Lyapunov functions for systems with structured uncertainties*, Automatica, 39 (2003), pp. 1027–1035.
[13] M. C. de Oliveira, J. C. Geromel, and L. Hsu, *LMI characterization of structural and robust stability: The discrete-time case*, Linear Algebra Appl., 296 (1999), pp. 27–38.
[14] A. C. Enthoven and K. J. Arrow, *A theorem on expectations and the stability of equilibrium*, Econometrica, 24 (1956), pp. 288–293.
[15] R. Fleming, G. Grosman, T. Lenker, S. Narayan, and S.-C. Ong, *Classes of Schur D-stable matrices*, Linear Algebra Appl., 306 (2000), pp. 15–24.
[16] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *LMI Control Toolbox for Use with Matlab*, User's Guide, The MathWorks Inc., Natick, MA, 1995.
[17] J. C. Geromel, *Optimal linear filtering under parameter uncertainty*, IEEE Trans. Signal Process., 47 (1999), pp. 168–175.
[18] J. C. Geromel, M. C. de Oliveira, and L. Hsu, *LMI characterization of structural and robust stability*, Linear Algebra Appl., 285 (1998), pp. 69–80.

[19] J. C. Geromel, P. L. D. Peres, and J. Bernussou, *On a convex parameter space method for linear control design of uncertain systems*, SIAM J. Control Optim., 29 (1991), pp. 381–402.
[20] B. S. Goh, *Nonvulnerability of ecosystems in unpredictable environments*, Theoret. Population Biology, 10 (1976), pp. 83–95.
[21] B. S. Goh, *Global stability in many-species systems*, Amer. Naturalist, 111 (1977), pp. 135–143.
[22] D. Hershkowitz, *Recent directions in matrix stability*, Linear Algebra Appl., 171 (1992), pp. 161–186.
[23] C. R. Johnson, *Sufficient conditions for D-stability*, J. Econom. Theory, 9 (1974), pp. 53–62.
[24] W. S. Kafri, *Robust D-stability*, Appl. Math. Lett., 15 (2002), pp. 7–10.
[25] I. Kaminer, P. P. Khargonekar, and M. A. Rotea, *Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control for discrete-time systems via convex optimization*, Automatica, 29 (1993), pp. 57–70.
[26] G. V. Kanovei and D. O. Logofet, *D-stability of 4-by-4 matrices*, Comput. Math. Math. Phys., 38 (1998), pp. 1369–1374.
[27] E. Kaszkurewicz and A. Bhaya, *Robust stability and diagonal Liapunov functions*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 508–520.
[28] E. Kaszkurewicz and A. Bhaya, *Matrix Diagonal Stability in Systems and Computation*, Birkhäuser Boston, Boston, MA, 1999.
[29] E. Kaszkurewicz and L. Hsu, *A note on the absolute stability of nonlinear discrete time systems*, Internat. J. Control, 40 (1984), pp. 867–869.
[30] H. K. Khalil, *Nonlinear Systems*, Prentice-Hall, Upper Saddle River, NJ, 1996.
[31] H. K. Khalil and P. V. Kokotovic, *D-stability and multi-parameter singular perturbation*, SIAM J. Control Optim., 17 (1979), pp. 56–65.
[32] P. P. Khargonekar and M. A. Rotea, *Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control: A convex optimization approach*, IEEE Trans. Automat. Control, 36 (1991), pp. 824–837.
[33] J. Lee and T. F. Edgar, *Real structured singular value conditions for the strong D-stability*, Systems Control Lett., 44 (2001), pp. 273–277.
[34] R. M. Palhares and P. L. D. Peres, *Robust $\mathcal{H}_\infty$ filtering design with pole constraints for discrete-time systems: An LMI approach*, in Proceedings of the 1999 American Control Conference, Vol. 1, San Diego, CA, 1999, pp. 4418–4422.
[35] R. M. Palhares and P. L. D. Peres, *Robust $\mathcal{H}_\infty$ filtering design with pole placement constraint via LMIs*, J. Optim. Theory Appl., 102 (1999), pp. 239–261.
[36] P. A. Parrilo and S. Lall, *Semidefinite programming relaxations and algebraic optimization in control*, European J. Control, 9 (2003), pp. 307–321.
[37] S. K. Persidskii, *Problem of absolute stability*, Automat. Remote Control, 12 (1969), pp. 1889–1895.
[38] J. Quirk and R. Ruppert, *Qualitative economics and the stability of equilibrium*, Rev. Econom. Stud., 32 (1965), pp. 311–326.
[39] D. C. W. Ramos and P. L. D. Peres, *A less conservative LMI condition for the robust stability of discrete-time uncertain systems*, Systems Control Lett., 43 (2001), pp. 371–378.
[40] D. C. W. Ramos and P. L. D. Peres, *An LMI condition for the robust stability of uncertain continuous-time linear systems*, IEEE Trans. Automat. Control, 47 (2002), pp. 675–678.
[41] R. Redheffer, *Volterra multipliers.* I, SIAM J. Algebraic Discrete Methods, 6 (1985), pp. 592–611.
[42] C. W. Scherer, *Higher-order relaxations for robust LMI problems with verifications for exactness*, in Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, HI, 2003, pp. 4652–4657.
[43] D. D. Šiljak, *Large Scale Dynamic Systems: Stability and Structure*, North-Holland, Amsterdam, 1978.
[44] J. F. Sturm, *Using SeDuMi* 1.02: *A MATLAB toolbox for optimization over symmetric cones*, Optim. Methods Softw., 11/12 (1999), pp. 625–653. Software available online from http://fewcal.kub.nl/sturm/software/sedumi.html.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 2, pp. 426–440

UPPER AND LOWER BOUNDS FOR THE TAILS OF THE DISTRIBUTION OF THE CONDITION NUMBER OF A GAUSSIAN MATRIX*

JEAN-MARC AZAÏS† AND MARIO WSCHEBOR‡

Abstract. Let A be an $m\times m$ real random matrix with independently and identically distributed standard Gaussian entries. We prove that there exist universal positive constants c and C such that the tail of the probability distribution of the condition number $\kappa(A)$ satisfies the inequalities $\frac{c}{x} < \mathrm{P}\{\kappa(A) > mx\} < \frac{C}{x}$ for every $x > 1$. The proof requires a new estimation of the joint density of the largest and the smallest eigenvalues of A^TA which follows from a formula for the expectation of the number of zeros of a certain random field defined on a smooth manifold.

Key words. random matrices, condition number, eigenvalue distribution, Rice formulae

AMS subject classifications. 15A12, 60G15

DOI. 10.1137/S0895479803429764

1. Introduction and main result. Let A be an $m \times m$ real matrix and denote by

$$\|A\| = \sup_{\|x\|=1} \|Ax\|$$

its Euclidean operator norm. $\|x\|$ denotes the Euclidean norm of x in $\mathbb{R}^m$. If A is nonsingular, its *condition number* $\kappa(A)$ is defined by

$$\kappa(A) = \|A\|\|A^{-1}\|$$

(von Neumann and Goldstine [18]; Turing [17]). The role of $\kappa(A)$ in a variety of numerical analysis problems is well established (see, for example, Wilkinson [20], Smale [14], Higham [9], and Demmel [6]). The purpose of the present paper is to prove the following.

THEOREM 1.1. *Assume that $A = ((a_{ij}))_{i,j=1,\dots,m}$, $m \geq 3$, and that the a_{ij}'s are independently and identically distributed (i.i.d.) Gaussian standard random variables.*

Then there exist universal positive constants c, C such that for $x > 1$,

$$\frac{c}{x} < \mathrm{P}\{\kappa(A) > mx\} < \frac{C}{x}. \tag{1.1}$$

Remarks. The following are remarks on the statement of Theorem 1.1:

1. It is well known that as m tends to infinity, the distribution of the random variable $\kappa(A)/m$ converges to a certain distribution (this follows easily, for example, from Edelman [7]). The interest of (1.1) lies in the fact that it holds true for all $m \geq 3$ and $x > 1$

*Received by the editors June 10, 2003; accepted for publication (in revised form) April 26, 2004; published electronically January 12, 2005. This work was supported by ECOS program U03E01.
http://www.siam.org/journals/simax/26-2/42976.html

†Laboratoire de Statistique et Probabilités, UMR-CNRS C55830, Université Paul Sabatier, 118, route de Narbonne, 31062 Toulouse Cedex 4, France (azais@cict.fr).

‡Centro de Matemática, Facultad de Ciencias, Universidad de la República, Calle Igua 4225, 11400 Montevideo, Uruguay (wschebor@cmat.edu.uy).

2. We will see below that $c = 0.13$, $C = 5.60$ satisfy (1.1) for every $m = 3, 4, \ldots$ and $x > 1$. Using the same methods, one can obtain more precise upper and lower bounds for each m, but we will not detail these calculations here.
   The simulation study of section 3 suggests that $\mathrm{P}\{\kappa(A) > mx\}$ is increasing with m, so that c should be the value corresponding to $m = 3$, i.e., $c \approx 0.88$, and C the one derived from the mentioned asymptotic result in Edelman [7], i.e., $C = 2$.
3. In Sankar, Spielman, and Teng [13] it was conjectured that

$$\mathrm{P}\{\kappa(A) > x\} = O\left(\frac{m}{\sigma x}\right)$$

   when the a_{ij}'s are independent Gaussian random variables having a common variance $\sigma^2 \leq 1$ and $\sup_{i,j} |\mathrm{E}(a_{ij})| \leq 1$.
   The upper bound part of (1.1) implies that this conjecture holds true in the centered case. The lower bound shows that, up to a constant factor, this is the exact order of the behavior of the tail of the probability distribution of $\kappa(A)$. See Wschebor [22] for the noncentered case.
4. This theorem, and related ones, can be considered as results on the Wishart matrix $A^T A$ (A^T denotes the transpose of A). Introducing some minor changes, it is possible to use the same methods to study the condition number of $A^T A$ for rectangular $n \times m$ matrices A having i.i.d. Gaussian standard entries, $n > m$. This will be considered elsewhere.

Some examples of related results on the random variable $\kappa(A)$ are the following.

THEOREM 1.2 (see [7]). *Under the same hypothesis as that of Theorem* 1.1, *one has*

$$E(\log \kappa(A)) = \log m + C_1 + \varepsilon_m,$$

where C_1 is a known constant ($C_1 \approx 1.537$) and $\varepsilon_m \to 0$ as $m \to +\infty$.

THEOREM 1.3 (see [4]). *Let $A = ((a_{ij}))_{i,j=1,\ldots,m}$ and assume that the a_{ij}'s are independent Gaussian random variables with a common variance σ^2 and $m_{ij} = \mathrm{E}(a_{ij})$. Denote by $M = ((m_{ij}))_{i,j=1,\ldots,m}$ the nonrandom matrix of expectations. Then*

$$\mathrm{E}(\log \kappa(A)) \leq \log m + \log\left[\frac{\|M\|}{\sigma\sqrt{m}} + 4\right] + C_1',$$

where C_1' is a known constant.

Next, we introduce some notation. Given A, an $m \times m$ real matrix, we denote by $\lambda_1, \ldots, \lambda_m$, $0 \leq \lambda_1 \leq \cdots \leq \lambda_m$, the eigenvalues of $A^T A$. If $X : S^{m-1} \to \mathbb{R}$ is the quadratic polynomial $X(x) = x^T A^T A x$, then

- $\lambda_m = \|A\|^2 = \max_{x \in S^{m-1}} X(x)$,
- in case $\lambda_1 > 0$, $\lambda_1 = \frac{1}{\|A^{-1}\|^2} = \min_{x \in S^{m-1}} X(x)$.

It follows that

$$\kappa(A) = \left(\frac{\lambda_m}{\lambda_1}\right)^{\frac{1}{2}}$$

when $\lambda_1 > 0$. We put $\kappa(A) = +\infty$ if $\lambda_1 = 0$. Note also that $\kappa(A) \geq 1$ and $\kappa(rA) = \kappa(A)$ for any real r, $r \neq 0$.

There is an important difference between the proof of Theorem 1.1 and those of the two other theorems mentioned above. In the latter cases, one puts

$$\log \kappa(A) = \frac{1}{2}\log \lambda_m - \frac{1}{2}\log \lambda_1,$$

and if one takes expectations, the joint distribution of the random variables λ_m, λ_1 does not play any role; the proof uses only the individual distributions of λ_m and λ_1. On the contrary, the proof below of Theorem 1.1 depends essentially on the joint distribution of the pair (λ_m, λ_1). A general formula for the joint density of $\lambda_1, \dots, \lambda_m$ has been well known for a long time (see, for example, Wilks [21], Wigner [19], Krishnaiah and Chang [11], Kendall, Stuart, and Ord [10], and the references therein), but it seems to be difficult to adapt this to our present requirements. In fact, we will use a different approach, based on the expected value of the number of zeros of a random field parameterized on a smooth manifold.

We have also applied this technique to give a new proof of the known result of Lemma 2.2, a lower bound for $\mathrm{P}\{\lambda_1 < a\}$.

One can ask if Theorem 1.1 follows from the well-known exponential bounds for the concentration of the distribution of λ_m together with known bounds for the distribution of λ_1 (see, for example, Szarek [15], Davidson and Szarek [5], and Ledoux [12] for these types of inequalities).

More precisely, consider the upper bound in Theorem 1.1. For $\varepsilon > 0$ one has

$$\begin{aligned}
\mathrm{P}\big\{\kappa(A) > mx\big\} &= \mathrm{P}\left\{\frac{\lambda_m}{\lambda_1} > m^2x^2\right\} \\
&\le \mathrm{P}\big\{\lambda_m > (4+\varepsilon)m\big\} + \mathrm{P}\left\{\lambda_m \le (4+\varepsilon)m, \frac{\lambda_m}{\lambda_1} > m^2x^2\right\} \\
&\le \mathrm{P}\big\{\lambda_m > (4+\varepsilon)m\big\} + \mathrm{P}\left\{\lambda_1 < \frac{(4+\varepsilon)}{mx^2}\right\} \\
&\le C_1 \exp\big[-C_2 m\varepsilon^2\big] + C_3\frac{\sqrt{4+\varepsilon}}{x}, \qquad (1.2)
\end{aligned}$$

where C_1, C_2, C_3 are positive constants. From (1.2), making an adequate choice of ε one can get an upper bound for $\mathrm{P}\{\kappa(A) > mx\}$ of the form $(\text{const})\frac{1}{x}\big(\frac{\log x}{m}\big)^{\alpha}$ for some $\alpha > 0$ and x large enough. However, this kind of argument does not lead to the precise order given by our Theorem 1.1.

On the other hand, using known results for the distribution of other functions of the spectrum (for example, $(\lambda_1 + \cdots + \lambda_m)/\lambda_1$ as in Edelman [8]), one can get upper and lower bounds for the tails of the distribution of $\kappa(A)$ which again do not reach the precise behavior $(\text{const})/x$.

2. Proof of Theorem 1.1. It is easy to see that, almost surely, the eigenvalues of A^TA are pairwise different. We introduce the following additional notation:

- $\langle .,. \rangle$ is usual scalar product in $\mathbb{R}^m$ and $\{e_1, \dots, e_m\}$ the canonical basis.
- I_k denotes the $k \times k$ identity matrix.
- $B = A^TA = ((b_{ij}))_{i,j=1,\dots,m}$.
- For $s \neq 0$ in $\mathbb{R}^m$, $\pi_s : \mathbb{R}^m \to \mathbb{R}^m$ denotes the orthogonal projection onto $\{s\}^{\perp}$, the orthogonal complement of s in $\mathbb{R}^m$.
- $M \succ 0$ (resp., $M \prec 0$) means that the symmetric matrix M is positive definite (resp., negative definite).

- If ξ is a random vector, $p_\xi(.)$ is the density of its distribution whenever it exists.
- For a differentiable function F defined on a smooth manifold M embedded in some Euclidean space, $F'(s)$ and $F''(s)$ are the first and the second derivative of F that we will represent, in each case, with respect to an appropriate orthonormal basis of the tangent space.

Instead of (1.1) we prove the equivalent statement: for $x > m$,

$$\frac{cm}{x} < \mathrm{P}\{\kappa(A) > x\} < \frac{Cm}{x}. \tag{2.1}$$

We break the proof into several steps. Our main task is to estimate the joint density of the pair (λ_m, λ_1); this will be done in Step 4.

Step 1. For $a, b \in \mathbb{R}$, $a > b$, one has almost surely

$$\begin{aligned}\{\lambda_m \in (a, a+da)\,, \lambda_1 \in (b, b+db)\} \\ = \left\{ \begin{array}{c} \exists\ s,t \in S^{m-1}, \langle s,t\rangle = 0, X(s) \in (a, a+da)\,, X(t) \in (b, b+db)\,, \\ \pi_s(Bs) = 0, \pi_t(Bt) = 0, X''(s) \prec 0, X''(t) \succ 0 \end{array} \right\}.\end{aligned} \tag{2.2}$$

An instant reflection shows that almost surely the number

$$N_{a,b,da,db}$$

of pairs (s,t) belonging to the right-hand side of (2.2) is equal to 0 or to 4, so that

$$\mathrm{P}\{\lambda_m \in (a, a+da), \lambda_1 \in (b, b+db)\} = \frac{1}{4}\mathrm{E}(N_{a,b,da,db}). \tag{2.3}$$

Step 2. In this step we will give a bound for $\mathrm{E}\,(N_{a,b,da,db})$ using what we call a Rice-type formula (see Azaïs and Wschebor [3] for some related problems and general tools). Let

$$V = \left\{(s,t) : s,t \in S^{m-1}, \langle s,t\rangle = 0\right\}.$$

V is a C^∞-differentiable manifold without boundary, embedded in $\mathbb{R}^{2m}$, $\dim(V) = 2m-3$. We will denote by $\tau = (s,t)$ a generic point in V and by $\sigma_V(d\tau)$ the geometric measure on V.

It is easy to see that $\sigma_V(V) = \sqrt{2}\sigma_{m-1}.\sigma_{m-2}$, where σ_{m-1} denotes the surface area of $S^{m-1} \subset \mathbb{R}^m$, that is, $\sigma_{m-1} = \frac{2\pi^{m/2}}{\Gamma(m/2)}$. On V we define the random field

$$Y : V \to \mathbb{R}^{2m}$$

by means of

$$Y(s,t) = \left(\begin{array}{c} \pi_s(Bs) \\ \pi_t(Bt) \end{array} \right).$$

For $\tau = (s,t)$ a given point in V, we have that

$$Y(\tau) \in \{(t,-s)\}^\perp \cap \left\{\{s\}^\perp \times \{t\}^\perp\right\} = W_\tau$$

for any value of the matrix B, where $\{(t,-s)\}^\perp$ is the orthogonal complement of the point $(t,-s)$ in $\mathbb{R}^{2m}$.

In fact, $(t,-s) \in \{s\}^{\perp} \times \{t\}^{\perp}$ and

$$\begin{aligned}\langle Y(s,t),(t,-s)\rangle_{\mathbb{R}^{2m}} &= \langle \pi_s(Bs),t\rangle - \langle \pi_t(Bt),s\rangle \\ &= \langle Bs - \langle s,Bs\rangle s,t\rangle - \langle Bt - \langle t,Bt\rangle t,s\rangle = 0\end{aligned}$$

since $\langle s,t\rangle = 0$ and B is symmetric. Notice that $\dim(W_\tau) = 2m-3$.

We also set

$$\Delta(\tau) = \left[\det\left[(Y'(\tau))^T\, Y'(\tau)\right]\right]^{\frac{1}{2}},$$

$$N = \sharp\left\{\tau : \tau \in V, Y(\tau) = 0\right\}.$$

For $\tau = (s,t) \in V$, F_τ denotes the event

$$F_\tau = \{X(s) \in (a,a+da), X(t) \in (b,b+db), X''(s) \prec 0, X''(t) \succ 0\},$$

and $p_{Y(\tau)}(.)$ is the density of the random vector $Y(\tau)$ in the $(2m-3)$-dimensional subspace W_τ of $\mathbb{R}^{2m}$.

Assume that 0 is not a critical value of Y, that is, if $Y(\tau) = 0$, then $\Delta(\tau) \neq 0$. This holds true with probability 1. By compactness of V, this implies $N < \infty$. Assume that $N \neq 0$ and denote by $\tau_1, \ldots, \tau_N$ the roots of the equation $Y(\tau) = 0$.

Because of the implicit function theorem, if $\delta > 0$ is small enough, one can find in V open neighborhoods $U_1, \ldots, U_N$ of the points $\tau_1, \ldots, \tau_N$, respectively, so that the following hold:

- Y is a diffeomorphism between U_j and $Y(V) \cap B_{2m}(0,\delta)$ ($B_{2m}(0,\delta)$ is the Euclidean ball of radius δ centered at the origin, in $\mathbb{R}^{2m}$).
- $U_1, \ldots, U_N$ are pairwise disjoint.
- If $\tau \notin \bigcup_{j=1}^N U_j$, then $Y(\tau) \notin B_{2m}(0,\delta)$.

Using the change of variable formula, it follows that

$$\int_V \Delta(\tau)\ 1_{\{\|Y(\tau)\|<\delta\}} \sigma_V(d\tau) = \sum_{j=1}^N \int_{U_j} \Delta(\tau)\ \sigma_V(d\tau) = \sum_{j=1}^N \mu\left(Y(U_j)\right), \tag{2.4}$$

where $\mu\left(Y(U_j)\right)$ denotes the—$(2m-3)$-dimensional—geometric measure of $Y(U_j)$. As $\delta \downarrow 0$, $\mu\left(Y(U_j)\right) \sim |B_{2m-3}(\delta)|$, where $|B_{2m-3}(\delta)|$ is the $(2m-3)$-dimensional Lebesgue measure of a ball of radius δ in $\mathbb{R}^{2m-3}$. It follows from (2.4) that, almost surely,

$$N = \lim_{\delta \downarrow 0} \frac{1}{|B_{2m-3}(\delta)|} \int_V \Delta(\tau) \mathbb{1}_{\{\|Y(\tau)\|<\delta\}} \sigma_V(d\tau).$$

In exactly the same way, one can prove that

$$N_{a,b,da,db} = \lim_{\delta \downarrow 0} \frac{1}{|B_{2m-3}(\delta)|} \int_V \Delta(\tau) \mathbb{1}_{F_\tau} \mathbb{1}_{\{\|Y(\tau)\|<\delta\}} \sigma_V(d\tau).$$

Applying Fatou's lemma and Fubini's theorem,

$$\begin{aligned}\mathrm{E}(N_{a,b,da,db}) &\leq \liminf_{\delta \downarrow 0} \frac{1}{|B_{2m-3}(\delta)|} \int_V \mathrm{E}\left(\Delta(\tau)\ \mathbb{1}_{F_\tau} \mathbb{1}_{\{\|Y(\tau)\|<\delta\}}\right) \sigma_V(d\tau) \\ &= \liminf_{\delta \downarrow 0} \int_V \sigma_V(d\tau) \int_{B_{m,\delta,\tau}} \mathrm{E}\left(\Delta(\tau) \mathbb{1}_{F_\tau} / Y(\tau) = y\right) p_{Y(\tau)}(y)\ \frac{dy}{|B_{2m-3}(\delta)|} \\ &= \int_V \mathrm{E}\left(\Delta(\tau) \mathbb{1}_{F_\tau} / Y(\tau) = 0\right) p_{Y(\tau)}(0)\ \sigma_V(d\tau),\end{aligned}$$

where $B_{m,\delta,\tau} = B_{2m}(0,\delta) \cap W_\tau$. The validity of the last passage to the limit will become clear below, since it will follow from the calculations we will perform that the integrand in the inner integral is a continuous function of the pair (τ, y). Hence,

$$\mathrm{E}\,(N_{a,b,da,db}) \le \int_a^{a+da} dx \int_b^{b+db} dy \int_V \mathrm{E}\,\big(\Delta(s,t)\mathbb{1}_{\{X''(s)\prec 0, X''(t)\succ 0\}}/\mathcal{C}_{s,t,x,y}\big) \times p_{X(s),X(t),Y(s,t)}(x,y,0)\ \sigma_V(d(s,t)), \tag{2.5}$$

where $\mathcal{C}_{s,t,x,y}$ is the condition $\{X(s) = x, X(t) = y, Y(s,t) = 0\}$. The invariance of the law of A with respect to isometries of $\mathbb{R}^m$ implies that the integrand in (2.5) does not depend on $(s,t) \in V$. Hence, we have proved that the joint law of λ_m and λ_1 has a density $g(a,b)$, $a > b$, and

$$g(a,b) \le \frac{\sqrt{2}}{4}\sigma_{m-1}\sigma_{m-2}\mathrm{E}\,\big(\Delta(e_1,e_2)\mathbb{1}_{\{X''(e_1)\prec 0, X''(e_2)\succ 0\}}/\mathcal{C}_{e_1,e_2,a,b}\big) \times p_{X(e_1),X(e_2),Y(e_1,e_2)}(a,b,0). \tag{2.6}$$

In fact, using the method of Azaïs and Wschebor [3], it could be proved that (2.6) is an equality, but we do not need such a precise result here.

Step 3. Next, we compute the ingredients in the right-hand member of (2.6). We take as orthonormal basis for the subspace $W_{(e_1,e_2)}$

$$\left\{(e_3,0),\dots,(e_m,0),(0,e_3),\dots,(0,e_m),\frac{1}{\sqrt{2}}(e_2,e_1)\right\} = L_1.$$

We have

$$\begin{aligned} X(e_1) &= b_{11},\\ X(e_2) &= b_{22},\\ X''(e_1) &= B_1 - b_{11}I_{m-1},\\ X''(e_2) &= B_2 - b_{22}I_{m-1}, \end{aligned}$$

where B_1 (resp., B_2) is the $(m-1)\times(m-1)$ matrix obtained by suppressing the first (resp., the second) row and column in B,

$$Y(e_1,e_2) = (0, b_{21}, b_{31},\dots,b_{m1},b_{12},0,b_{32},\dots,b_{m2})^T,$$

so that it has the following expression in the orthonormal basis L_1:

$$Y(e_1,e_2) = \sum_{i=3}^m \big(b_{i1}(e_i,0) + b_{i2}(0,e_i)\big) + \sqrt{2}b_{12}\left(\frac{1}{\sqrt{2}}(e_2,e_1)\right).$$

It follows that the joint density of $X(e_1), X(e_2), Y(e_1,e_2)$ appearing in (2.6) in the space $\mathbb{R}\times\mathbb{R}\times W_{(e_1,e_2)}$ is the joint density of the random variables

$$b_{11}, b_{22}, \sqrt{2}b_{12}, b_{31},\dots,b_{m1},b_{32},\dots,b_{m2}$$

at the point $(a,b,0,\dots,0)$. To compute this density, first compute the joint density q of

$$b_{31},\dots,b_{m1},b_{32},\dots,b_{m2},$$

given a_1, a_2, where a_j denotes the jth column of A which is Gaussian standard in $\mathbb{R}^m$. q is the normal density in $\mathbb{R}^{2(m-2)}$, centered with variance matrix

$$\begin{pmatrix} \|a_1\|^2 I_{m-2} & \langle a_1, a_2\rangle I_{m-2} \\ \langle a_1, a_2\rangle I_{m-2} & \|a_2\|^2 I_{m-2} \end{pmatrix}.$$

Set

$$a_j' = \frac{a_j}{\|a_j\|},\ j = 1, 2.$$

The density of the triplet

$$(b_{11}, b_{22}, b_{12}) = (\|a_1\|^2, \|a_2\|^2, \|a_1\|\|a_2\|\langle a_1', a_2'\rangle)$$

at the point $(a, b, 0)$ can be computed as follows.

Since $\langle a_1', a_2'\rangle$ and $(\|a_1\|, \|a_2\|)$ are independent, the density of the triplet at $(a, b, 0)$ is equal to

$$\chi^2_m(a)\chi^2_m(b)(ab)^{-1/2} p_{<a_1', a_2'>}(0),$$

where $\chi^2_m(.)$ denotes the χ^2 density with m degrees of freedom.

Let $\xi = (\xi_1, \dots, \xi_m)^T$ be Gaussian standard in $\mathbb{R}^m$. Clearly, $\langle a_1', a_2'\rangle$ has the same distribution as $\frac{\xi_1}{\|\xi\|}$, because of the invariance under rotations.

$$\begin{aligned}\frac{1}{2t}\mathrm{P}\{|\langle a_1', a_2'\rangle| \le t\} = \frac{1}{2t}\mathrm{P}\left\{\frac{\xi_1^2}{\chi^2_{m-1}} \le \frac{t^2}{1-t^2}\right\} &= \frac{1}{2t}\mathrm{P}\left\{F_{1,m-1} \le \frac{t^2(m-1)}{1-t^2}\right\} \\ &= \frac{1}{2t}\int_0^{\frac{t^2(m-1)}{1-t^2}} f_{1,m-1}(x)dx,\end{aligned}$$

where $\chi^2_{m-1} = \xi_2^2 + \cdots + \xi_m^2$ and $F_{1,m-1}$ has the Fisher distribution with $(1, m-1)$ degrees of freedom and density $f_{1,m-1}$. Letting $t \to 0$, we obtain

$$p_{\langle a_1', a_2'\rangle}(0) = \frac{1}{\sqrt{\pi}}\frac{\Gamma(m/2)}{\Gamma\big((m-1)/2\big)}.$$

Summing up, the density in (2.6) is equal to

$$\frac{1}{\sqrt{2}}(2\pi)^{2-m}\pi^{-\frac{1}{2}}\frac{1}{\Gamma(m/2)\Gamma\big((m-1)/2\big)}2^{-m}\frac{1}{\sqrt{ab}}\exp\left(-\frac{a+b}{2}\right). \tag{2.7}$$

We now consider the conditional expectation in (2.6). First, observe that the $(2m-3)$-dimensional tangent space to V at the point (s, t) is parallel to the orthogonal complement in $\mathbb{R}^m \times \mathbb{R}^m$ of the triplet of vectors $(s, 0); (0, t); (t, s)$. This is immediate from the definition of V.

To compute the associated matrix for $Y'(e_1, e_2)$ take the set

$$\left\{(e_3, 0), \dots, (e_m, 0), (0, e_3), \dots, (0, e_m), \frac{1}{\sqrt{2}}(e_2, -e_1)\right\} = L_2$$

as orthonormal basis in the tangent space and the canonical basis in $\mathbb{R}^{2m}$. A direct calculation gives

$$Y'(e_1,e_2)=\begin{pmatrix} -v^T & 0_{1,m-2} & -\frac{1}{\sqrt{2}}b_{21} \\ w^T & 0_{1,m-2} & \frac{1}{\sqrt{2}}(-b_{11}+b_{22}) \\ B_{12}-b_{11}I_{m-2} & 0_{m-2,m-2} & \frac{1}{\sqrt{2}}w \\ 0_{1,m-2} & -w^T & \frac{1}{\sqrt{2}}(-b_{11}+b_{22}) \\ 0_{1,m-2} & v^T & \frac{1}{\sqrt{2}}b_{21} \\ 0_{m-2,m-2} & B_{12}-b_{22}I_{m-2} & -\frac{1}{\sqrt{2}}v \end{pmatrix},$$

where $v^T=(b_{31},\dots,b_{m1}), w^T=(b_{32},\dots,b_{m2}), 0_{i,j}$ is a null matrix with i rows and j columns, and B_{12} is obtained from B by suppressing the first and second rows and columns. The columns represent the derivatives in the directions of L_2 at the point (e_1,e_2). The first m rows correspond to the components of $\pi_s(Bs)$, the last m ones to those of $\pi_t(Bt)$. Thus, under the condition $\mathcal{C}_{e_1,e_2,a,b}$ that is used in (2.6),

$$Y'(e_1,e_2)=\begin{pmatrix} 0_{1,m-2} & 0_{1,m-2} & 0 \\ 0_{1,m-2} & 0_{1,m-2} & \frac{1}{\sqrt{2}}(b-a) \\ B_{12}-aI_{m-2} & 0_{m-2,m-2} & 0_{m-2,1} \\ 0_{1,m-2} & 0_{1,m-2} & \frac{1}{\sqrt{2}}(b-a) \\ 0_{1,m-2} & 0_{1,m-2} & 0 \\ 0_{m-2,m-2} & B_{12}-bI_{m-2} & 0_{m-2,1} \end{pmatrix}$$

and

$$\Big[\det\big[\big(Y'(e_1,e_2)\big)^TY'(e_1,e_2)\big]\Big]^{\frac{1}{2}}=|\det(B_{12}-aI_{m-2})||\det(B_{12}-bI_{m-2})|(a-b).$$

Step 4. Note that $B_1-aI_{m-1}\prec 0\Rightarrow B_{12}-aI_{m-2}\prec 0$, and similarly, $B_2-bI_{m-1}\succ 0\Rightarrow B_{12}-bI_{m-2}\succ 0$, and that for $a>b$, under $\mathcal{C}_{e_1,e_2,a,b}$, there is equivalence in these relations.

It is also clear that, since $B_{12}\succ 0$, one has

$$|\det(B_{12}-aI_{m-2})|\mathbb{1}_{B_{12}-aI_{m-2}\prec 0}\le a^{m-2},$$

and it follows that the conditional expectation in (2.6) is bounded by

$$a^{m-1}\mathrm{E}\big(|\det(B_{12}-bI_{m-2})|\mathbb{1}_{B_{12}-bI_{m-2}\succ 0}/\mathcal{C}\big), \tag{2.8}$$

where $\mathcal{C}$ is the condition $\{b_{11}=a,b_{22}=b,b_{12}=0,b_{i1}=b_{i2}=0\ (i=3,\dots,m)\}$.

To compute the conditional expectation in (2.8) we further condition on the value of the random vectors a_1 and a_2. Since unconditionally $a_3,\dots,a_m$ are i.i.d. standard Gaussian vectors in $\mathbb{R}^m$, under this new conditioning, their joint law becomes the law of i.i.d. standard Gaussian vectors in $\mathbb{R}^{m-2}$ and independent of the condition. That is, (2.8) is equal to

$$a^{m-1}\mathrm{E}\big(|\det(M-bI_{m-2})|\mathbb{1}_{M-bI_{m-2}\succ 0}\big), \tag{2.9}$$

where M is an $(m-2)\times(m-2)$ random matrix with entries $M_{ij}=\langle v_i,v_j\rangle$ $(i,j=1,\dots,m-2)$ and the vectors $v_1,\dots,v_{m-2}$ are i.i.d. Gaussian standard in $\mathbb{R}^{m-2}$. The expression in (2.9) is bounded by

$$a^{m-1}\mathrm{E}\big(\det(M)\big)=a^{m-1}(m-2)!.$$

The last equality is contained in the following lemma, which is well known; see, for example, Edelman [7].

LEMMA 2.1. *Let $\xi_1, \dots, \xi_m$ be i.i.d. random vectors in $\mathbb{R}^p$, $p \geq m$, their common distribution being Gaussian centered with variance I_p.*

Denote by $W_{m,p}$ the matrix

$$W_{m,p} = ((\langle \xi_i, \xi_j \rangle))_{i,j=1,\dots,m},$$

and by

$$D(\lambda) = \det(W_{m,p} - \lambda I_m)$$

its characteristic polynomial.

Then

(i)

$$\mathrm{E}(\det(W_{m,p})) = p(p-1)\dots(p-m+1), \tag{2.10}$$

(ii)

$$\mathrm{E}(D(\lambda)) = \sum_{k=0}^{m} (-1)^k \binom{m}{k} \frac{p!}{(p-m+k)!} \lambda^k. \tag{2.11}$$

Returning to the proof of the theorem and summing up this part, after substituting in (2.6), we get

$$g(a,b) \leq C_m \frac{\exp\big(-(a+b)/2\big)}{\sqrt{ab}} a^{m-1}, \tag{2.12}$$

where $C_m = \frac{1}{4(m-2)!}$.

Step 5. Now we prove the upper-bound part in (2.1). One has, for $x > 1$,

$$\mathrm{P}\{\kappa(A) > x\} = \mathrm{P}\left\{\frac{\lambda_m}{\lambda_1} > x^2\right\} \leq \mathrm{P}\left\{\lambda_1 < \frac{L^2 m}{x^2}\right\} + \mathrm{P}\left\{\frac{\lambda_m}{\lambda_1} > x^2, \lambda_1 \geq \frac{L^2 m}{x^2}\right\}, \tag{2.13}$$

where L is a positive number to be chosen later on. For the first term in (2.13), we use Proposition 9 in Cuesta-Albertos and Wschebor [4], which is a slight modification of Theorem 3.2 in Sankar, Spielman, and Teng [13]:

$$\mathrm{P}\left\{\lambda_1 < \frac{L^2 m}{x^2}\right\} = \mathrm{P}\left\{\|A^{-1}\| > \frac{x}{L\sqrt{m}}\right\} \leq C_2(m) \frac{Lm}{x}.$$

Here,

$$C_2(m) = \left(\frac{2}{\pi}\right)^{\frac{1}{2}} \left[\sup_{0<c<m} \sqrt{c}\,\mathrm{P}\left\{t_{m-1}^2 > \frac{(m-1)c}{m-c}\right\}\right]^{-1} \leq C_2(+\infty) \approx 2.3473,$$

where t_{m-1} is a random variable having Student's distribution with $m-1$ degrees of freedom.

For the second term in (2.13),

$$\mathrm{P}\left\{\frac{\lambda_m}{\lambda_1} > x^2, \lambda_1 \geq \frac{L^2 m}{x^2}\right\} = \int_{L^2 m x^{-2}}^{+\infty} db \int_{bx^2}^{+\infty} g(a,b)\,da \leq G_m(x^2)$$

with

$$G_m(y) = C_m \int_{L^2 m y^{-1}}^{+\infty} db \int_{by}^{+\infty} \frac{\exp\big(-(a+b)/2\big)}{\sqrt{ab}} a^{m-1} da,$$

using (2.12). We have

$$G'_m(y) = C_m \Big[-\int_{L^2 m y^{-1}}^{+\infty} \exp(-b/2)\sqrt{b}\exp\big(-(by)/2\big)(by)^{m-3/2} db + L^2 m y^{-2} \int_{L^2 m}^{+\infty} \exp\Big(-\frac{1}{2}\Big(a + \frac{L^2 m}{y}\Big)\Big) a^{m-3/2} L^{-1} m^{-\frac{1}{2}} y^{\frac{1}{2}} da \Big], \tag{2.14}$$

which implies

$$\begin{aligned} -G'_m(y) &\le C_m y^{m-3/2} \int_{L^2 m y^{-1}}^{+\infty} \exp\Big(-\frac{b(1+y)}{2}\Big) b^{m-1} db \\ &= \frac{y^{-3/2}}{4(m-2)!} \Big(\frac{y}{1+y}\Big)^m 2^m \int_{\frac{L^2 m}{2y}(1+y)}^{+\infty} e^{-z} z^{m-1} dz \\ &\le \frac{y^{-3/2}}{4(m-2)!} 2^m \int_{\frac{L^2 m}{2}}^{+\infty} e^{-z} z^{m-1} dz. \end{aligned}$$

Put $I_m(a) = \int_a^{+\infty} e^{-z} z^{m-1} dz$. Integrating by parts,

$$I_m(a) = e^{-a}\big[a^{m-1} + (m-1)a^{m-2} + (m-1)(m-2)a^{m-3} + \cdots + (m-1)!\big],$$

so that for $a > 2.5m$

$$I_m(a) \le \frac{5}{3} e^{-a} a^{m-1}.$$

If $L^2 > 5$, we obtain the bound

$$-G'_m(y) \le D_m y^{-3/2} \quad \text{with} \quad D_m = \frac{5}{6} \frac{m^{m-1}}{(m-2)!} L^{2(m-1)} \exp\Big(-\frac{L^2 m}{2}\Big).$$

We now apply Stirling's formula (Abramowitz and Stegun [1, sect. 6.1.38]), i.e., for all $x > 0$

$$\Gamma(x+1)\exp\Big(-\frac{1}{12x}\Big) \le \Big(\frac{x}{e}\Big)^x \sqrt{2\pi x} \le \Gamma(x+1),$$

to get

$$D_m \le \frac{5\sqrt{2}}{12\sqrt{\pi} L^2} \frac{m}{\sqrt{m-2}} \exp\Big(-m\frac{L^2 - 4log(L) - 2}{2}\Big) \le \frac{5\sqrt{2}}{12\sqrt{\pi} L^2} m,$$

if we choose for L the only root larger than 1 of the equation $L^2 - 4\log(L) - 2 = 0$ (check that $L \approx 2.3145$). To finish,

$$0 \le G_m(y) = \int_y^{+\infty} -G'_m(t) dt < D_m \int_y^{+\infty} \frac{dt}{t^{3/2}} = 2 D_m y^{-\frac{1}{2}}.$$

Replacing y by x^2 and performing the numerical evaluations, the upper bound in (2.1) follows, and we get for the constant C the value 5.60.

Step 6. We consider now the lower bound in (2.1). For $\gamma > 0$ and $x > 1$, we have

$$\begin{aligned}(2.15)\quad \mathrm{P}\{\kappa(A) > x\} &= \mathrm{P}\left\{\frac{\lambda_m}{\lambda_1} > x^2\right\} \geq \mathrm{P}\left\{\frac{\lambda_m}{\lambda_1} > x^2, \lambda_1 < \frac{\gamma^2 m}{x^2}\right\} \\ &= \mathrm{P}\left\{\lambda_1 < \frac{\gamma^2 m}{x^2}\right\} - \mathrm{P}\left\{\frac{\lambda_m}{\lambda_1} \leq x^2, \lambda_1 < \frac{\gamma^2 m}{x^2}\right\}.\end{aligned}$$

A lower bound for the first term in the right-hand member of (2.15) is obtained using the following inequality, which we state as a separate lemma. In fact, this result is known; see, for example, Szarek [15, Theorem 1.2], where it is proved without giving an explicit value for the constant. See also Edelman [7, Corollary 3.1], for a related result.

LEMMA 2.2. *If* $0 < a < 1/m$, *then*

$$\mathrm{P}\{\lambda_1 < a\} \geq \beta\sqrt{am},$$

where we can choose $\beta = \left(\frac{2}{3}\right)^{3/2} e^{-1/3}$.

Proof. Define the index $i_X(t)$ of a critical point $t \in S^{m-1}$ of the function X as the number of negative eigenvalues of $X"(t)$. For each $a > 0$ put

$$N_i(a) = \sharp\{t \in S^{m-1} : X(t) = t^T B t < a, X'(t) = 0, i_X(t) = i\}$$

for $i = 0, 1, \ldots, m-1$. One easily checks that if the eigenvalues of B are $\lambda_1, \ldots, \lambda_m$, $0 < \lambda_1 < \cdots < \lambda_m$, then

$$\begin{aligned}&\bullet \text{ if } a \leq \lambda_1, \text{ then } N_i(a) = 0 \\ &\qquad\qquad\qquad\qquad \text{for } i = 0, 1, \ldots, m-1; \\ &\bullet \text{ if } \lambda_i < a \leq \lambda_{i+1}, \text{ then } N_k(a) = 2 \\ &\text{for some } i = 0, 1, \ldots, m_1 \qquad \text{for } k = 0, \ldots, i-1, \\ &\qquad\qquad\qquad\qquad N_k(a) = 0 \\ &\qquad\qquad\qquad\qquad \text{for } k = i, \ldots, m-1; \\ &\bullet \text{ if } \lambda_m < a, \text{ then } N_i(a) = 2 \\ &\qquad\qquad\qquad\qquad \text{for } i = 0, 1, \ldots, m-1.\end{aligned}$$

Now consider

$$M(a) = \sum_{i=0}^{m-1} (-1)^i N_i(a).$$

$M(a)$ is the Euler characteristic of the set $S = \{t \in S^{m-1} : X(t) < a\}$; see Adler [2]. It follows from the relations above that

- if $N_0(a) = 0$, then $N_i(a) = 0$ for $i = 1, \ldots, m-1$, and hence $M(a) = 0$;
- if $N_0(a) = 2$, then $M(a) = 0$ or 2,

so that in any case

$$M(a) \leq N_0(a).$$

Hence,

$$\mathrm{P}\{\lambda_1 < a\} = \mathrm{P}\{N_0(a) = 2\} = \frac{1}{2}\mathrm{E}\big(N_0(a)\big) \geq \frac{1}{2}\mathrm{E}\big(M(a)\big). \tag{2.16}$$

The expectation of $M(a)$ can be written using the Rice-type formula (see Azaïs and Wschebor [3] or Taylor and Adler [16])

$$\begin{aligned}\mathrm{E}\big(M(a)\big) &= \int_0^a dy \int_{S^{m-1}} \mathrm{E}\big[\det\big(X"(t)\big)/X(t) = y, X'(t) = 0\big] p_{X(t),X'(t)}(y,0)\sigma_{m-1}(dt) \\ &= \int_0^a \sigma_{m-1}(S^{m-1})\mathrm{E}\big[\det\big(X"(e_1)\big)/X(e_1) = y, X'(e_1) = 0\big] p_{X(e_1),X'(e_1)}(y,0)dy,\end{aligned}$$

where we have used again invariance under isometries. Applying a similar Gaussian regression—as we did in Step 4 to get rid of the conditioning—we obtain

$$\mathrm{E}\big(M(a)\big) = \int_0^a \mathrm{E}\big[\det\big(Q - yI_{m-1}\big)\big] \frac{\sqrt{2\pi}}{2^{m-1}}\Gamma^{-2}\left(\frac{m}{2}\right)\frac{\exp(-y/2)}{\sqrt{y}}dy, \tag{2.17}$$

where Q is an $(m-1)\times(m-1)$ random matrix with entry i,j equal to $(\langle v_i, v_j\rangle)$ and $v_1,\dots,v_{m-1}$ are i.i.d. Gaussian standard in $\mathbb{R}^{m-1}$. We now use part (ii) of Lemma 2.1:

$$\mathrm{E}\big[\det\big(Q - yI_{m-1}\big)\big] = (m-1)!\sum_{k=0}^{m-1}\left(\begin{array}{c} m-1 \\ k \end{array}\right)\frac{(-y)^k}{k!}. \tag{2.18}$$

Under condition $0 < a < m^{-1}$, since $0 < y < a$, as k increases, the terms of the sum in the right-hand member of (2.18) have decreasing absolute value, so that

$$\mathrm{E}\big[\det\big(Q - yI_{m-1}\big)\big] \geq (m-1)![1-(m-1)y].$$

Substituting into the right-hand member of (2.17), we get

$$\mathrm{E}\big[M(a)\big] \geq \frac{\sqrt{2\pi}}{2^{m-1}}\frac{(m-1)!}{\Gamma^2(m/2)}J_m(a),$$

where, using again $0 < a < m^{-1}$,

$$J_m(a) = \int_0^a \big(1-(m-1)y\big)\frac{\exp(-y/2)}{\sqrt{y}}dy \geq \int_0^a \frac{\big(1-(m-1)y\big)}{\sqrt{y}}(1-y/2)dy \geq \frac{4}{3}\sqrt{a}$$

by an elementary computation. Going back to (2.17), applying Stirling's formula, and remarking that $\big(1+1/n\big)^{n+1} \geq e$, we get

$$\mathrm{P}\{\lambda_1 < a\} \geq \left(\frac{2}{3}\right)^{3/2} e^{-1/3}\sqrt{am}.$$

This proves the lemma. □

End of the proof of Theorem 1.1. Using Lemma 2.2, the first term on the right-hand side of (2.15) is bounded below by

$$\beta\gamma\frac{m}{x}.$$

TABLE 1

Values of the estimations $\mathrm{P}\{\kappa(A) > mx\}$ *for* $x = 1, 2, 3, 5, 10, 15, 30, 50, 100$ *and* $m = 3, 5, 10, 30, 100, 300, 500$ *by Monte Carlo method over* 40,000 *simulations.*

	Value of x								
Probability	1	2	3	5	10	20	30	50	100
Lower bound: $.13/x$	.13	.065	.043	.026	.013	.007	.004	.003	.001
Upper bound: $5.6/x$	1	1	1	1	.56	.28	.187	.112	.056
$m = 3$	.881	.57	.41	.26	.13	.067	.044	.027	.013
$m = 5$	.931	.66	.48	.30	.16	.079	.053	.033	.016
$m = 10$	.959	.71	.52	.34	.17	.088	.059	.035	.017
$m = 30$	.974	.75	.56	.36	.19	.096	.063	.038	.019
$m = 100$	.978	.77	.58	.38	.20	.098	.066	.040	.019
$m = 300$	.982	.77	.58	.38	.20	.101	.069	.041	.022
$m = 500$	.980	.77	.59	.38	.20	.100	.066	.039	.020

To obtain a bound for the second term, we use again our upper bound (2.12) on the joint density $g(a, b)$, so that we obtain

$$\begin{aligned}(2.19)\quad \mathrm{P}\Big\{\frac{\lambda_m}{\lambda_1} \le x^2, \lambda_1 < \frac{\gamma^2 m}{x^2}\Big\} &= \int_0^{\frac{\gamma^2 m}{x^2}} db \int_b^{bx^2} g(a,b)da \\ &\le C_m \int_0^{\frac{\gamma^2 m}{x^2}} db \int_b^{bx^2} \frac{\exp\big(-(a+b)/2\big)}{\sqrt{ab}} a^{m-1} da \\ &\le C_m \int_0^{\frac{\gamma^2 m}{x^2}} b(x^2-1) b^{-\frac{1}{2}} (bx^2)^{\frac{m-3}{2}} db \\ &\le \frac{1}{4(m-2)!} \frac{x^2-1}{x^3} \gamma^{2m} m^{m-1} \le \frac{\sqrt{2}}{8\sqrt{\pi}} e^m \gamma^{2m} \frac{m}{x}\end{aligned}$$

on applying Stirling's formula. Now choosing $\gamma = 1/e$, we see that the hypothesis of Lemma 2.2 is satisfied and also

$$\mathrm{P}\Big\{\frac{\lambda_m}{\lambda_1} \le x^2, \lambda_1 < \frac{\gamma^2 m}{x^2}\Big\} \le \frac{\sqrt{2}}{8\sqrt{\pi}} e^{-3} \frac{m}{x}.$$

Substituting into (2.15), we obtain the lower bound in (1.1) with

$$c = \left(\frac{2}{3}\right)^{3/2} e^{-4/3} - \frac{\sqrt{2}}{8\sqrt{\pi}} e^{-3} \approx 0.138. \qquad \square$$

3. Monte Carlo experiment. To study the tail of the distribution of the condition number of Gaussian matrices of various size, we used the following Matlab functions:

- `normrnd`, to simulate normal variables;
- `cond`, to compute the condition number of matrix A.

The results of over 40,000 simulations using Matlab are given in Table 1 and in Figure 1.

The table suggests, taking into account the simulation variability, that the constants c and C should take values smaller than 0.88 and bigger than 2.00, respectively.

Acknowledgments. The authors thank Professors G. Letac and F. Cucker for valuable discussions. They also thank the associate editor and two anonymous referees for helpful comments that contributed to improving this paper.

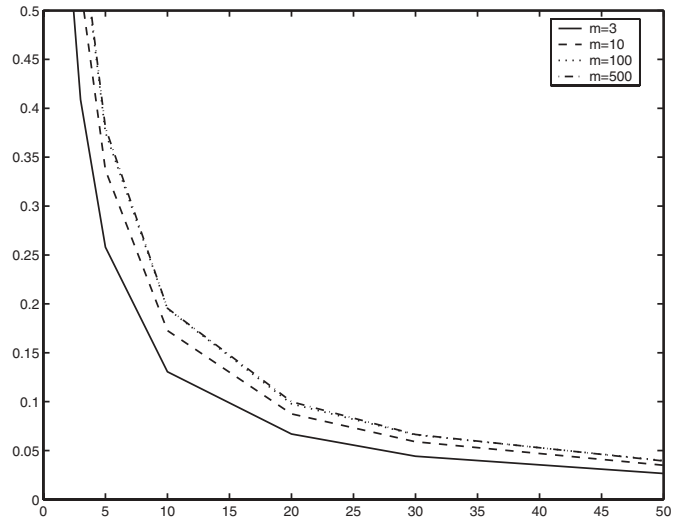


FIG. 1. *Values of* $\mathrm{P}\{\kappa(A) > mx\}$ *as a function of* x *for* $m = 3, 10, 100,$ *and* 500.

REFERENCES

[1] M. ABRAMOWITZ AND I. A. STEGUN, EDS., *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, John Wiley & Sons, New York, 1984

[2] R. J. ADLER, *The Geometry of Random Fields*, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, Chichester, UK, 1981.

[3] J.-M. AZAÏS AND M. WSCHEBOR, *On the distribution of the maximum of a Gaussian field with d parameters*, Ann. Appl. Probab., to appear.

[4] J. CUESTA-ALBERTOS AND M. WSCHEBOR, *Condition numbers and extrema of random fields*, in Seminar on Stochastic Analysis, Random Fields and Applications IV, Progr. Probab. 58, Birkhäuser, Basel, 2004, pp. 69–82.

[5] K. R. DAVIDSON AND S. J. SZAREK, *Local operator theory, random matrices and Banach spaces*, in Handbook of the Geometry of Banach Spaces, Vol. I, North-Holland, Amsterdam, 2001, pp. 317–366.

[6] J. W. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[7] A. EDELMAN, *Eigenvalues and condition numbers of random matrices*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 543–560.

[8] A. EDELMAN, *On the distribution of a scaled condition number*, Math. Comp., 58 (1992), pp. 185–190.

[9] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[10] M. KENDALL, A. STUART, AND J. K. ORD, *The Advanced Theory of Statistics. Vol.* 3, 4th ed., Macmillan, New York, 1983.

[11] P. R. KRISHNAIAH AND T. C. CHANG, *On the exact distributions of the extreme roots of the Wishart and* MANOVA *matrices*, J. Multivariate Anal., 1 (1971), pp. 108–117.

[12] M. LEDOUX, *A remark on hypercontractivity and tail inequalities for the largest eigenvalues of random matrices*, Séminaire de Probabilités XXXVII, Lecture Notes in Math. 1832, Springer-Verlag, Berlin, 2003, pp. 360–369.

[13] A. SANKAR, D. A. SPIELMAN, AND S. H. TENG, *Smoothed Analysis of the Condition Number and Growth Factors of Matrices*, preprint, 2002.

[14] S. SMALE, *On the efficiency of algorithms of analysis*, Bull. Amer. Math. Soc. (N.S.), 13 (1985), pp. 87–121.

[15] S. J. SZAREK, *Condition numbers of random matrices*, J. Complexity, 7 (1991), pp. 131–149.

[16] J. E. TAYLOR AND R. J. ADLER, *Euler characteristics for Gaussian fields on manifolds*, Ann. Probab., 31 (2003), pp. 533–563.

[17] A. M. TURING, *Rounding-off errors in matrix processes*, Quart. J. Mech. Appl. Math., 1 (1948), pp. 287–308.

[18] J. von Neumann and H. H. Goldstine, *Numerical inverting of matrices of high order*, Bull. Amer. Math. Soc., 53 (1947), pp. 1021–1099.
[19] E. P. Wigner, *Random matrices in physics*, SIAM Rev., 9 (1967), pp. 1–23.
[20] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, NJ, 1963.
[21] S. S. Wilks, *Mathematical Statistics*, John Wiley & Sons, New York, 1962.
[22] M. Wschebor, *Smoothed analysis of* $\kappa(a)$, J. Complexity, 20 (2004), pp. 97–107.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 2, pp. 441–456

EIGENVALUES AND CONDITION NUMBERS OF COMPLEX RANDOM MATRICES*

T. RATNARAJAH†, R. VAILLANCOURT†, AND M. ALVO†

Abstract. In this paper, the distributions of the largest and smallest eigenvalues of complex Wishart matrices and the condition number of complex Gaussian random matrices are derived. These distributions are represented by complex hypergeometric functions of matrix arguments, which can be expressed in terms of complex zonal polynomials. Several results are derived on complex hypergeometric functions and complex zonal polynomials and are used to evaluate these distributions. Finally, applications of these distributions in numerical analysis and statistical hypothesis testing are mentioned.

Key words. complex random matrix, complex Wishart matrix, complex zonal polynomials, eigenvalue distribution, condition number distribution

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

DOI. 10.1137/S089547980342204X

1. Introduction.

In this work, we investigate the distributions of the eigenvalues and condition number of complex random matrices and their applications to numerical analysis. In contrast to [3], we consider that the elements of random matrices are complex Gaussian distributed with zero mean and arbitrary covariance matrices. This will enable us to consider the beautiful but difficult theory of complex zonal polynomials (also called Schur polynomials [10]), which are symmetric polynomials in the eigenvalues of a complex matrix [12]. Complex zonal polynomials enable us to represent the distributions of the eigenvalues of these complex random matrices as infinite series.

In statistics, the random eigenvalues are used in hypothesis testing, principal component analysis, canonical correlation analysis, multiple discriminant analysis, etc. (see [12]). In nuclear physics, random eigenvalues are used to model nuclear energy levels and level spacing [11]. Moreover, the zeros of the Riemann zeta function are modeled using random eigenvalues [11].

Let an $n\times m$ complex Gaussian random matrix $\mathbf{A}$ be distributed as $\mathbf{A} \sim \mathcal{C}N(\mathbf{0}, I_n \otimes \Sigma)$ with mean $\mathcal{E}\{\mathbf{A}\} = \mathbf{0}$ and covariance $\mathrm{cov}\{\mathbf{A}\} = I_n \otimes \Sigma$. Then the matrix $\mathbf{W} = \mathbf{A}^H\mathbf{A}$ is called the complex central Wishart matrix, and its distribution is denoted by $\mathcal{C}W_m(n, \Sigma)$.

The condition number, $\mathrm{cond}(A)$, of a matrix A is defined as the positive square root of the ratio of the largest to the smallest eigenvalues of the positive definite Hermitian matrix $W = A^H A$. Thus (see [4] and [19])

$$\mathrm{cond}(A) = \sqrt{\lambda_{\max}/\lambda_{\min}} = \|A\|_2\|A^{-1}\|_2, \qquad \mathrm{cond}(W) = \mathrm{cond}(A)^2, \tag{1.1}$$

*Received by the editors July 24, 2003; accepted for publication (in revised form) by D. Boley April 27, 2004; published electronically January 12, 2005. This work was partially supported by the Natural Sciences and Engineering Council of Canada.

http://www.siam.org/journals/simax/26-2/42204.html

†Department of Mathematics and Statistics, University of Ottawa, 585 King Edward Ave., Ottawa ON K1N 6N5, Canada (t.ratnarajah@ieee.org, remi@uottawa.ca, malvo@science.uottawa.ca). The first author is now with ECIT, Queen's University of Belfast, Belfast BT7 1NN, Northern Ireland, UK. The second author is the corresponding author.

where the ℓ_2-norms of the matrix A and the vector x are

$$\|A\|_2 = \sup_{x \neq 0} \|Ax\|_2 / \|x\|_2 \quad \text{and} \quad \|x\|_2 = \left(x_1^2 + x_2^2 + \cdots + x_n^2\right)^{1/2},$$

respectively. We assume that the eigenvalues of W are ordered in strictly decreasing order, $\lambda_{\max} = \lambda_1 > \cdots > \lambda_m = \lambda_{\min} > 0$, since the probability that any eigenvalues of A are equal is zero. The condition number of a random matrix gives valuable information on the convergence rate of iterative methods in optimization algorithms and on the reliability of the solutions of linear systems of equations.

The distributions of $\lambda_{\max}$ and $\lambda_{\min}$ and the condition number density of random matrices are studied in [3] (and references given therein) for $\Sigma = I$ (see also [18]). The singular value distribution of Gaussian random matrices is given in [13] for $\Sigma = I$. Note that the singular values of a complex Gaussian random matrix A are equal to the square root of the eigenvalues of the complex Wishart matrix $W = A^H A$. The asymptotic distribution of the largest eigenvalue of a complex Wishart matrix is given in [6] if m and n are large and $\Sigma = I$. In [7] the largest and smallest eigenvalue distributions of a complex Wishart matrix are studied for $\Sigma = \sigma^2 I$. Here, we derive the distributions of the largest and smallest eigenvalues of complex Wishart matrices and the condition number density of complex random matrices for arbitrary Σ. Applications of these distributions are also given.

This paper is organized as follows. Section 2 provides the necessary tools for deriving the eigenvalue and condition number distributions of complex central Wishart matrices. Complex central Wishart matrices are studied in section 3 and their largest and smallest eigenvalue distributions derived. The condition number density is derived in section 4 and a numerical example is given.

2. Preliminaries. In this section, we derive several results on complex hypergeometric functions and complex zonal polynomials that will be used to evaluate the subsequent distributions. First, we define the multivariate hypergeometric coefficients $[a]_\kappa^{(\alpha)}$ which frequently occur in integrals involving zonal polynomials. Let $\kappa = (k_1, \ldots, k_m)$ be a partition of the integer k with $k_1 \geq \cdots \geq k_m \geq 0$ and $k = k_1 + \cdots + k_m$. Then [1]

$$[a]_\kappa^{(\alpha)} = \prod_{i=1}^{m} \left(a - \frac{1}{\alpha}(i-1)\right)_{k_i},$$

where $(a)_k = a(a+1)\cdots(a+k-1)$ and $\alpha = 1$ for complex and $\alpha = 2$ for real multivariate hypergeometric coefficients, respectively. In this paper we consider only the complex case; therefore, for notational simplicity we drop the superscript [8], i.e.,

$$[a]_\kappa := [a]_\kappa^{(1)} = \prod_{i=1}^{m} (a - i + 1)_{k_i} = \frac{\mathcal{C}\Gamma_m(a, \kappa)}{\mathcal{C}\Gamma_m(a)},$$

where

$$\mathcal{C}\Gamma_m(a, \kappa) = \pi^{m(m-1)/2} \prod_{i=1}^{m} \Gamma(a + k_i - i + 1), \quad \Re(a) > (m-1),$$

and $\mathcal{C}\Gamma_m(a)$ denotes the complex multivariate gamma function

$$\mathcal{C}\Gamma_m(a) = \pi^{m(m-1)/2} \prod_{k=1}^{m} \Gamma(a - k + 1), \quad \Re(a) > (m-1) + k_1.$$

Moreover,

$$\mathcal{C}\Gamma_m(a,-\kappa) = \pi^{m(m-1)/2}\prod_{i=1}^{m}\Gamma(a-m-k_i+i).$$

The complex zonal polynomials (also called Schur polynomials [10]) of a complex matrix X are defined in [5] by

$$C_\kappa(X) = \chi_{[\kappa]}(1)\chi_{[\kappa]}(X), \tag{2.1}$$

where $\chi_{[\kappa]}(1)$ is the dimension of the representation $[\kappa]$ of the symmetric group given by

$$\chi_{[\kappa]}(1) = k!\,\frac{\prod_{i<j}^{m}(k_i-k_j-i+j)}{\prod_{i=1}^{m}(k_i+m-i)!}, \tag{2.2}$$

and $\chi_{[\kappa]}(X)$ is the character of the representation $[\kappa]$ of the linear group given as a symmetric function of the eigenvalues, $\lambda_1,\ldots,\lambda_m$, of X by

$$\chi_{[\kappa]}(X) = \frac{\det\left[\left(\lambda_i^{k_j+m-j}\right)\right]}{\det\left[\left(\lambda_i^{m-j}\right)\right]}. \tag{2.3}$$

Note that both the real and complex zonal polynomials are particular cases of Jack polynomials $C_\kappa^{(\alpha)}(X)$ for general α. See [1] and [15] for details. Again $\alpha = 1$ for complex and $\alpha = 2$ for real zonal polynomials, respectively. For the same reason as before, we shall drop the superscript of Jack polynomials, as we did in (2.1), i.e., $C_\kappa(X) := C_\kappa^{(1)}(X)$.

The following basic properties are given in [5]:

$$(\operatorname{tr} X)^k = \sum_\kappa C_\kappa(X)$$

and

$$\int_{U(m)} C_\kappa(AXBX^H)(dX) = \frac{C_\kappa(A)C_\kappa(B)}{C_\kappa(I_m)}, \tag{2.4}$$

where (dX) is the invariant measure on the unitary group $U(m)$ normalized to make the total measure unity and

$$C_\kappa(I_m) = 2^{2k}k!\left[\frac{1}{2}\,m\right]_\kappa \frac{\prod_{i<j}^{r}(2k_i-2k_j-i+j)}{\prod_{i=1}^{r}(2k_i+r-i)!},$$

where

$$\left[\frac{1}{2}m\right]_\kappa = \prod_{i=1}^{r}\left(\frac{1}{2}(m-i+1)\right)_{k_i}.$$

Note that the partition κ of k has r nonzero parts.

The probability distributions of random matrices are often derived in terms of hypergeometric functions of matrix arguments. The following definitions of hypergeometric functions with a single and double matrix argument are due to Constantine [2] and Baker [1].

DEFINITION 2.1. *The hypergeometric function of one complex matrix is defined as*

$$ {}_pF_q^{(\alpha)}(a_1,\ldots,a_p;b_1,\ldots,b_q;X) = \sum_{k=0}^{\infty}\sum_{\kappa}\frac{[a_1]_\kappa^{(\alpha)}\cdots[a_p]_\kappa^{(\alpha)}}{[b_1]_\kappa^{(\alpha)}\cdots[b_q]_\kappa^{(\alpha)}}\frac{C_\kappa^{(\alpha)}(X)}{k!}, \tag{2.5}$$

where $X \in \mathbb{C}^{m\times m}$ and $\{a_i\}_{i=1}^p$ and $\{b_i\}_{i=1}^q$ are arbitrary complex numbers. Note that $\sum_\kappa$ denotes summation over all partitions κ of k and $\alpha = 1$ and 2 for complex and real hypergeometric functions, respectively.

In this paper we consider only the complex case, and hence we shall drop the superscript, i.e., ${}_pF_q := {}_pF_q^{(1)}$. Note that none of the parameters b_i is allowed to be zero or an integer or half-integer $\le m-1$. Otherwise some of the terms in the denominator will be zero [12].

Remark 1. The convergence of (2.5) is as follows [12]:

(i) If $p \le q$, then the series converges for all X.
(ii) If $p = q+1$, then the series converges for $\sigma(X) < 1$, where the spectral radius $\sigma(X)$ of X is the maximum of the absolute values of the eigenvalues of X.
(iii) If $p > q+1$, then the series diverges for all $X \neq 0$, unless it terminates. Note that the series terminates when some of the numerators $[a_j]_\kappa$ in the series vanish.

Special cases are

$$ {}_0F_0(X) = \operatorname{etr}(X), \qquad {}_1F_0(a;X) = \det(I-X)^{-a}, $$

and

$$ {}_0F_1(n;ZZ^H) = \int_{U(n)} \operatorname{etr}(ZE + \overline{ZE})(dE), $$

where Z is an $m\times n$ complex matrix with $m \le n$, etr denotes the exponential of the trace, $\operatorname{etr}(\cdot) = \exp(\operatorname{tr}(\cdot))$, and $\overline{ZE}$ denotes the complex conjugate of ZE.

DEFINITION 2.2. *The complex hypergeometric function of two complex matrices is defined by*

$$ {}_pF_q(a_1,\ldots,a_p;b_1,\ldots,b_q;X,Y) = \sum_{k=0}^{\infty}\sum_{\kappa}\frac{[a_1]_\kappa\cdots[a_p]_\kappa}{[b_1]_\kappa\cdots[b_q]_\kappa}\frac{C_\kappa(X)C_\kappa(Y)}{k!C_\kappa(I_m)}, \tag{2.6}$$

where $X,\ Y \in \mathbb{C}^{m\times m}$.

The splitting formula is

$$ \int_{U(m)} {}_pF_q(AEBE^H)(dE) = {}_pF_q(A,B). $$

The following propositions and corollaries are required in what follows.

PROPOSITION 2.3. *If Y and Z are $m\times m$ Hermitian matrices with $\Re(Z) > 0$, then*

$$ \int_{X^H=X>0} \operatorname{etr}(-XZ)(\det X)^{a-m}C_\kappa(XY)(dX) = \mathcal{C}\Gamma_m(a,\kappa)(\det Z)^{-a}C_\kappa(YZ^{-1}), \tag{2.7}$$

where $\Re(a) > (m-1)$, *and*

$$\int_{X^H=X>0} \operatorname{etr}(-XZ)(\det X)^{a-m} C_\kappa(X^{-1}Y)(dX) = \mathcal{C}\Gamma_m(a,-\kappa)(\det Z)^{-a} C_\kappa(YZ), \tag{2.8}$$

where $\Re(a) > (m-1) + k_1$.

Proof. Let $Z = I$ and $f(Y)$ denote the left side of (2.7). Then

$$f(EYE^H) = \int_{X^H=X>0} \operatorname{etr}(-X)(\det X)^{a-m} C_\kappa(XEYE^H)(dX) \quad \forall\, E \in U(m).$$

If $X = EWE^H$, then $(dX) = (dW)$ and $f(EYE^H) = f(Y)$. This implies that f is a symmetric function of Y. Moreover, (dE) is the normalized invariant measure on the unitary group $U(m)$. Therefore we have

$$\begin{aligned} f(Y) &= \int_{U(m)} f(Y)(dE) \\ &= \int_{X^H=X>0} \operatorname{etr}(-X)(\det X)^{a-m} \int_{U(m)} C_\kappa(XEYE^H)(dE)(dX) \\ &= \int_{X^H=X>0} \operatorname{etr}(-X)(\det X)^{a-m} \frac{C_\kappa(X)C_\kappa(Y)}{C_\kappa(I_m)}(dX) \\ &= \frac{f(I_m)C_\kappa(Y)}{C_\kappa(I_m)}. \end{aligned} \tag{2.9}$$

On the one hand, from Definition 7.2.1 in [12] we have

$$f(Y) = \frac{f(I_m)}{C_\kappa(I_m)}\, d_\kappa y_1^{k_1} \cdots y_m^{k_m} + \text{terms of lower weight.} \tag{2.10}$$

On the other hand, using Lemma 7.2.6 in [12] we have

$$\begin{aligned} f(Y) &= \int_{X^H=X>0} \operatorname{etr}(-X)(\det X)^{a-m} C_\kappa(XY)(dX) \\ &= d_\kappa y_1^{k_1} \cdots y_m^{k_m} \int_{X^H=X>0} \operatorname{etr}(-X)(\det X)^{a-m} x_{11}^{k_1-k_2} \\ &\quad \times \det \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}^{k_2-k_3} \cdots \det X^{k_m}(dX) + \text{terms of lower weight.} \end{aligned}$$

Substituting $X = T^H T$ and evaluating this integral we obtain

$$f(Y) = d_\kappa y_1^{k_1} \cdots y_m^{k_m} \mathcal{C}\Gamma_m(a,\kappa) + \text{terms of lower weight.} \tag{2.11}$$

Equating the coefficients of $y_1^{k_1} \cdots y_m^{k_m}$ in (2.10) and (2.11) and using (2.9), we obtain

$$f(Y) = \mathcal{C}\Gamma_m(a,\kappa) C_\kappa(Y).$$

The rest of the proof for general Z can be obtained by substituting $X = Z^{-1/2} V Z^{-1/2}$. Similarly, we can prove the second part. □

The following corollary follows from the second part of Proposition 2.3 by letting $Y = I$.

COROLLARY 2.4. *Let Z be an $m \times m$ Hermitian matrix with $\Re(Z) > 0$. Then*

(2.12)
$$\int_{X^H=X>0} \operatorname{etr}(-XZ)(\det X)^{a-m} C_\kappa(X^{-1})(dX) = \frac{(-1)^k \mathcal{C}\Gamma_m(a)}{[-a+m]_\kappa} (\det Z)^{-a} C_\kappa(Z)$$

for $\Re(a) > k_1 + (m-1)$, where $\kappa = (k_1, \ldots, k_m)$.

Proof. The result follows by noting that

$$\mathcal{C}\Gamma_m(a, -\kappa) = \frac{(-1)^k \mathcal{C}\Gamma_m(a)}{[-a+m]_\kappa}. \qquad \square$$

PROPOSITION 2.5. *Let Y be an $m \times m$ symmetric matrix. Then the following are true:*

(2.13)
$$\int_{0<X<I_m} (\det X)^{a-m} \det(I_m - X)^{b-m} C_\kappa(XY)(dX) = \frac{\mathcal{C}\Gamma_m(a,\kappa)\mathcal{C}\Gamma_m(b)}{\mathcal{C}\Gamma_m(a+b,\kappa)} C_\kappa(Y)$$

for $\Re(a) > (m-1)$ and $\Re(b) > (m-1)$. Moreover,

(2.14)
$$\int_{0<X<I_m} (\det X)^{a-m} \det(I_m - X)^{b-m} C_\kappa(X^{-1}Y)(dX) = \frac{\mathcal{C}\Gamma_m(a,-\kappa)\mathcal{C}\Gamma_m(b)}{\mathcal{C}\Gamma_m(a+b,-\kappa)} C_\kappa(Y)$$

for $\Re(a) > (m-1) + k_1$ and $\Re(b) > (m-1)$.

Proof. As in the proof of Proposition 2.3, if $f(Y)$ denotes the left side of (2.13), then we have

$$f(Y) = f(EYE^H) \;\forall\; E \in U(m) \quad \text{and} \quad f(Y)C_\kappa(I_m) = f(I_m)C_\kappa(Y).$$

Letting $Z = I$ and $Y = I$ in (2.7) and then multiplying with $f(I_m)$, we obtain

$$\begin{aligned}
\mathcal{C}\Gamma_m(a+b,\kappa) f(I_m) &= \int_{W^H=W>0} \operatorname{etr}(-W)(\det W)^{a+b-m} f(W)(dW) \\
&= \int_{W^H=W>0} \operatorname{etr}(-W)(\det W)^{a+b-m} \int_{0<X<I_m} (\det X)^{a-m} \\
&\quad \times \det(I_m - X)^{b-m} C_\kappa(WX)(dX)(dW).
\end{aligned}$$

Let $X = W^{-1/2} U W^{-1/2}$. Then $(dX) = (\det W)^{-m}(dU)$ and

$$\begin{aligned}
\mathcal{C}\Gamma_m(a+b,\kappa) f(I_m) &= \int_{W^H=W>0} \operatorname{etr}(-W) \int_{0<U<W} (\det U)^{a-m} \\
&\quad \times \det(W-U)^{b-m} C_\kappa(U)(dU)(dW) \\
&= \int_{U^H=U>0} \operatorname{etr}(-U)(\det U)^{a-m} C_\kappa(U)(dU) \\
&\quad \times \int_{V^H=V>0} \operatorname{etr}(-V)(\det V)^{b-m}(dV) \quad (\text{letting } V = W-U) \\
&= \mathcal{C}\Gamma_m(a,\kappa) C_\kappa(I_m) \mathcal{C}\Gamma_m(b).
\end{aligned}$$

This completes the proof, i.e.,

$$f(I_m) = \frac{\mathcal{C}\Gamma_m(a,\kappa)\mathcal{C}\Gamma_m(b)}{\mathcal{C}\Gamma_m(a+b,\kappa)}\, C_\kappa(I_m).$$

Similarly, we can prove the second part. □

If $b = m$, then we have the following corollary.

COROLLARY 2.6. *If Y is an $m \times m$ Hermitian matrix, then*

$$\int_{0<X<I_m} (\det X)^{a-m} C_\kappa(XY)(dX) = \frac{\mathcal{C}\Gamma_m(a)\mathcal{C}\Gamma_m(m)}{\mathcal{C}\Gamma_m(a+m)} \frac{[a]_\kappa}{[a+m]_\kappa} C_\kappa(Y) \tag{2.15}$$

for $\Re(a) > (m-1)$.

Proof. The result follows by noting that

$$\mathcal{C}\Gamma_m(a,\kappa) = [a]_\kappa \mathcal{C}\Gamma_m(a). \qquad \square$$

3. The complex central Wishart matrix. In this section, we describe the complex central Wishart distribution and give the joint eigenvalue density of the complex central Wishart matrix. The largest and smallest eigenvalue distributions are derived in subsections 3.1 and 3.2, respectively. These distributions are used in the hypothesis testing of the structure of the covariance matrix Σ.

The definition of the complex central Wishart distribution is given by the following.

DEFINITION 3.1. *Let $\mathbf{W} = \mathbf{A}^H\mathbf{A}$, where the $n \times m$ matrix $\mathbf{A}$ is distributed as $\mathbf{A} \sim \mathcal{C}N(\mathbf{0}, I_n \otimes \Sigma)$. Then $\mathbf{W}$ is said to have the complex central Wishart distribution $\mathbf{W} \sim \mathcal{C}W_m(n,\Sigma)$ with n degrees of freedom and covariance matrix Σ.*

Let $\mathbf{W} \sim \mathcal{C}W_m(n,\Sigma)$ with $n \geq m$. Then the density of $\mathbf{W}$ is given by [5]

$$f(W) = \frac{1}{\mathcal{C}\Gamma_m(n)(\det\Sigma)^n} \operatorname{etr}\left(-\Sigma^{-1}W\right)(\det W)^{n-m}. \tag{3.1}$$

Moreover, $\mathbf{W}$ is an $m \times m$ positive definite Hermitian matrix with real eigenvalues. The joint density of the eigenvalues, $\lambda_1 > \cdots > \lambda_m > 0$, of $\mathbf{W}$ is

$$f(\Lambda) = \frac{\pi^{m(m-1)}(\det\Sigma)^{-n}}{\mathcal{C}\Gamma_m(m)\mathcal{C}\Gamma_m(n)} \prod_{k=1}^{m} \lambda_k^{n-m} \prod_{k<l}^{m} (\lambda_k - \lambda_l)^2 {}_0F_0\left(-\Lambda, \Sigma^{-1}\right), \tag{3.2}$$

where $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_m)$.

If $\mathbf{W} \sim \mathcal{C}W_m(n, \sigma^2 I_m)$ with $n \geq m$, then the joint density of its eigenvalues is

$$f(\Lambda) = \frac{\pi^{m(m-1)}(\sigma^2)^{-nm}}{\mathcal{C}\Gamma_m(m)\mathcal{C}\Gamma_m(n)} \prod_{k=1}^{m} \lambda_k^{n-m} \prod_{k<l}^{m} (\lambda_k - \lambda_l)^2 \exp\left(-\frac{1}{\sigma^2}\sum_{k=1}^{m}\lambda_k\right). \tag{3.3}$$

3.1. Distribution of $\lambda_{\max}$. In this subsection, we derive the distribution of the largest eigenvalue, $\lambda_{\max}$, of a central Wishart matrix and apply it to hypothesis testing. The following theorem is needed.

THEOREM 3.2. *Let $\mathbf{W} \sim \mathcal{C}W_m(n,\Sigma)$ $(n \geq m)$ and let Δ be an $m \times m$ positive definite matrix. Then the probability $P(W < \Delta)$ is given by*

$$P(W < \Delta) = \frac{\mathcal{C}\Gamma_m(m)}{\mathcal{C}\Gamma_m(n+m)} \frac{(\det\Delta)^n}{(\det\Sigma)^n} {}_1F_1(n; n+m; -\Sigma^{-1}\Delta), \tag{3.4}$$

where

$$ {}_1F_1(a;b;X) = \sum_{k=0}^{\infty}\sum_{\kappa}\frac{[a]_\kappa}{[b]_\kappa}\frac{C_\kappa(X)}{k!}. $$

Proof. Using the Wishart distribution (3.1) we can write $P(W<\Delta)$ as

$$ P(W<\Delta) = \frac{1}{\mathcal{C}\Gamma_m(n)(\det\Sigma)^n}\int_{0<W<\Delta}\operatorname{etr}(-\Sigma^{-1}W)(\det W)^{n-m}\,(dW). $$

The change of variable $W=\Delta^{1/2}X\Delta^{1/2}$ leads to the differential form $(dW)=(\det\Delta)^m(dX)$. Hence,

$$ \begin{aligned} P(W<\Delta) &= \frac{(\det\Delta)^n}{\mathcal{C}\Gamma_m(n)(\det\Sigma)^n}\int_{0<X<I}\operatorname{etr}(-\Delta^{1/2}\Sigma^{-1}\Delta^{1/2}X)(\det X)^{n-m}\,(dX)\\ &= \frac{(\det\Delta)^n}{\mathcal{C}\Gamma_m(n)(\det\Sigma)^n}\sum_{k=0}^{\infty}\sum_{\kappa}\frac{1}{k!}\int_0^I(\det X)^{n-m}C_\kappa(-\Delta^{1/2}\Sigma^{-1}\Delta^{1/2}X)\,(dX)\\ &= \frac{\mathcal{C}\Gamma_m(m)}{\mathcal{C}\Gamma_m(n+m)}\frac{(\det\Delta)^n}{(\det\Sigma)^n}\sum_{k=0}^{\infty}\sum_{\kappa}\frac{[n]_\kappa}{[n+m]_\kappa}\frac{C_\kappa(-\Sigma^{-1}\Delta)}{k!}\\ &= \frac{\mathcal{C}\Gamma_m(m)}{\mathcal{C}\Gamma_m(n+m)}\frac{(\det\Delta)^n}{(\det\Sigma)^n}{}_1F_1(n;n+m;-\Sigma^{-1}\Delta). \end{aligned} $$

Note that Corollary 2.6 is used in this proof. □

The following corollary follows from Theorem 3.2.

COROLLARY 3.3. *Let* $\mathbf{W}\sim\mathcal{C}W_m(n,\Sigma)$ $(n\geq m)$. *If* $\lambda_{\max}$ *is the largest eigenvalue of* W, *then its distribution is given by*

$$ P(\lambda_{\max}<x) = \frac{\mathcal{C}\Gamma_m(m)}{\mathcal{C}\Gamma_m(n+m)}\frac{x^{mn}}{(\det\Sigma)^n}\,{}_1F_1(n;n+m;-x\Sigma^{-1}). \tag{3.5} $$

The density of $\lambda_{\max}$ *is obtained by differentiating* (3.5) *with respect to* x.

Proof. The inequality $\lambda_{\max}<x$ is equivalent to $W<xI$. Therefore, the result follows by letting $\Delta=xI$ in Theorem 3.2. □

The distributional result in Corollary 3.3 can be used to test hypotheses about Σ using statistics which are functions of $\lambda_{\max}$. For example, consider the null hypothesis $H_0:\Sigma=I_m$. A test on the size of α based on the largest eigenvalue $\lambda_{\max}$ is to reject H_0 if $\lambda_{\max}>\lambda(\alpha,m,n)$, where $\lambda(\alpha,m,n)$ is the upper $100\,\alpha\%$ point of the distribution of $\lambda_{\max}$ when $\Sigma=I_m$, i.e., $P_{I_m}(\lambda_{\max}>\lambda(\alpha,m,n))=\alpha$. The power function of this test is given by

$$ \beta(\Sigma) = P_\Sigma(\lambda_{\max}>\lambda(\alpha,m,n)), $$

which depends on Σ only through its eigenvalues. The percentage points and power can be computed using the distribution function given in Corollary 3.3.

3.2. Distribution of $\lambda_{\min}$. In this subsection, we derive the distribution of the smallest eigenvalue, $\lambda_{\min}$, of a central Wishart matrix and use it to test the structure of the covariance matrix Σ, as explained in the previous subsection. In addition, the distribution of $\lambda_{\min}$ is useful in principal component analysis. Here it would be of interest to find out the number of eigenvalues which are significant in Σ. The following theorem is used to derive the distribution of $\lambda_{\min}$.

THEOREM 3.4. *Let $\mathbf{W} \sim \mathcal{C}W_m(n, \Sigma)$ $(n \geq m)$ and let Δ be an $m \times m$ positive definite matrix. Then the probability $P(W > \Delta)$ can be written as a finite series, i.e.,*

$$P(W > \Delta) = \operatorname{etr}\left(-\Sigma^{-1}\Delta\right) \sum_{k=0}^{m(n-m)} \widehat{\sum_{\kappa}} \frac{C_k\left(\Sigma^{-1}\Delta\right)}{k!}, \tag{3.6}$$

where $\widehat{\sum}_{\kappa}$ denotes summation over the partitions $\kappa = (k_1, \ldots, k_m)$ of k with $k_1 \leq n - m$.

Proof. Using the Wishart distribution (3.1) we can write the probability $P(W > \Delta)$ as

$$P(W > \Delta) = \frac{1}{\mathcal{C}\Gamma_m(n)(\det \Sigma)^n} \int_{W>\Delta} \operatorname{etr}\left(-\Sigma^{-1}W\right)(\det W)^{n-m}(dW). \tag{3.7}$$

The change of variable $W = \Delta^{1/2}(I + X)\Delta^{1/2}$ leads to the differential form $(dW) = (\det \Delta)^m(dX)$. Hence,

$$\begin{aligned}
P(W > \Delta) &= \frac{\operatorname{etr}(-\Sigma^{-1}\Delta)(\det \Delta)^n}{\mathcal{C}\Gamma_m(n)(\det \Sigma)^n} \\
&\quad \times \int_{X>0} \operatorname{etr}\left(-\Delta^{1/2}\Sigma^{-1}\Delta^{1/2}X\right)(\det X)^{n-m}(\det(I + X^{-1}))^{n-m}(dX) \\
&= \frac{\operatorname{etr}(-\Sigma^{-1}\Delta)(\det \Delta)^n}{\mathcal{C}\Gamma_m(n)(\det \Sigma)^n} \sum_{k=0}^{m(n-m)} \widehat{\sum_{\kappa}} \frac{[-(n-m)]_{\kappa}(-1)^k}{k!} \\
&\quad \times \int_{X>0} \operatorname{etr}\left(-\Delta^{1/2}\Sigma^{-1}\Delta^{1/2}X\right)(\det X)^{n-m}C_{\kappa}(X^{-1})(dX) \\
&= \operatorname{etr}(-\Sigma^{-1}\Delta) \sum_{k=0}^{m(n-m)} \widehat{\sum_{\kappa}} \frac{C_{\kappa}\left(\Sigma^{-1}\Delta\right)}{k!}.
\end{aligned}$$

In this proof we have used

$$\begin{aligned}
\det\left(I + X^{-1}\right)^{n-m} &= {}_1F_0(-(n-m); -X^{-1}) \\
&= \sum_{k=0}^{m(n-m)} \widehat{\sum_{\kappa}} \frac{[-(n-m)]_{\kappa}C_{\kappa}(X^{-1})(-1)^k}{k!}
\end{aligned}$$

and Corollary 2.4. Note that if any part of κ is greater than $(n-m)$, then $[-(n-m)]_{\kappa} = 0$. Therefore, the series for ${}_1F_0$ reduces to a finite series. □

The distribution of the smallest eigenvalue is given in the following corollary.

COROLLARY 3.5. *Let $\mathbf{W} \sim \mathcal{C}W_m(n, \Sigma)$. If $\lambda_{\min}$ is the smallest eigenvalue of $\mathbf{W}$, then*

$$P(\lambda_{\min} > x) = \operatorname{etr}\left(-x\Sigma^{-1}\right) \sum_{k=0}^{m(n-m)} \widehat{\sum_{\kappa}} \frac{C_{\kappa}\left(x\Sigma^{-1}\right)}{k!}, \tag{3.8}$$

where $\widehat{\sum}_{\kappa}$ denotes summation over the partitions $\kappa = (k_1, \ldots, k_m)$ of k with $k_1 \leq n - m$. The density of $\lambda_{\min}$ is obtained by differentiating (3.8) with respect to x and then changing the sign.

Proof. The inequality $\lambda_{\min} > x$ is equivalent to $W > xI$. Therefore, the result follows by letting $\Delta = xI$ in Theorem 3.4. □

As a numerical example, we compute the smallest eigenvalue distribution of the complex central Wishart matrix for $m = 2$, $n = 10$, and

$$\Sigma = \begin{bmatrix} 1 & 0.25 + 0.25i \\ 0.25 - 0.25i & 1 \end{bmatrix}.$$

The distribution is defined by

$$P(\lambda_{\min} < x) = 1 - P(\lambda_{\min} > x),$$

where $P(\lambda_{\min} > x)$ is given in (3.8). Let $F = P(\lambda_{\min} < x)$. Figure 3.1 shows this distribution of $\lambda_{\min}$.

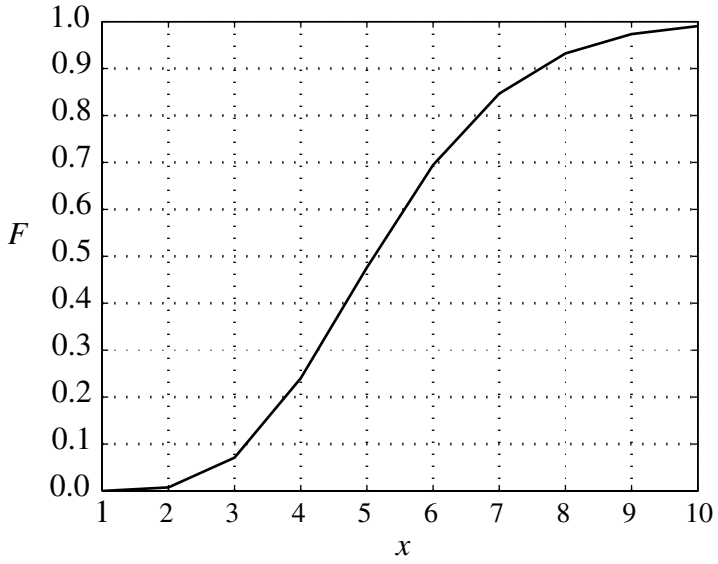


FIG. 3.1. *The smallest eigenvalue distribution of the complex central Wishart matrix.*

4. Distribution of cond(A). Many scientific problems lead to solving a random system of linear equations. The condition number distribution of this random matrix indicates how many digits of numerical precision are lost due to ill conditioning. In addition, if a random system is solved by an iterative technique, then the condition number distribution describes the speed of convergence of this iterative method (e.g., conjugate gradient method).

The condition number, cond(A), can also be defined (see [14], [22], and [3]) as the smallest number

$$\frac{\|\delta x\|}{\|x\|} \leq \text{cond}(A) \frac{\|\delta b\|}{\|b\|}$$

for all x and δx such that $Ax = b$ and $A(x + \delta x) = b + \delta b$. By taking the logarithm on both sides, we have

$$(\log \|\delta x\| - \log \|x\|) - (\log \|\delta b\| - \log \|b\|) \leq \log \text{cond}(A).$$

This shows that the number of correct digits in x can differ from the number of correct digits in b by at most $\log \operatorname{cond}(A)$. In [14], the loss of precision is denoted by $\log \operatorname{cond}(A)$. Problems where $\operatorname{cond}(A)$ is large are referred to as ill conditioned, and such problems are characterized by very elongated elliptical level sets. Iterative methods converge slowly for these problems. These facts illustrate the importance of the condition number distribution for solving random systems.

If $\mathbf{A} \sim \mathcal{C}N(0, I \otimes \Sigma)$ or $\mathbf{A} \sim \mathcal{C}N(0, I \otimes \sigma^2 I)$, then the condition number distributions of A and $W = A^H A$ are not available in the literature. We derive these distributions in what follows. First, we derive the joint density of the extreme eigenvalues of the complex central Wishart matrix $W = A^H A$, i.e., $f(\lambda_{\max}, \lambda_{\min})$. This will enable us to compute the distribution of the condition number of the random matrix A. The following two lemmas are required in what follows.

LEMMA 4.1. *Let* $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_m)$ *and* $D_1 = \{1 > \lambda_1 > \cdots > \lambda_m > 0\}$. *Then*

$$\int_{D_1} (\det \Lambda)^{a-m} \det(I - \Lambda)^{b-m} \prod_{k<l}^{m} (\lambda_k - \lambda_l)^2 C_\kappa(\Lambda) \bigwedge_{k=1}^{m} d\lambda_k = \frac{\mathcal{C}\Gamma_m(m)}{\pi^{m(m-1)}} \frac{\mathcal{C}\Gamma_m(a,\kappa)\mathcal{C}\Gamma_m(b)}{\mathcal{C}\Gamma_m(a+b,\kappa)} C_\kappa(I). \tag{4.1}$$

Proof. The result follows by letting $Y = I$ and $X = E\Lambda E^H$ in (2.13) and using the differential form

$$(dX) = \prod_{k<l}^{m} (\lambda_k - \lambda_l)^2 (d\Lambda)(E^H dE) \quad \text{with} \quad \int_{U(m)} (E^H dE) = \frac{2^m \pi^{m^2}}{\mathcal{C}\Gamma_m(m)}.$$

We must then divide the left side of (2.13) by $(2\pi)^m$. □

LEMMA 4.2. *Let* $Z = \operatorname{diag}(\zeta_2, \ldots, \zeta_m)$, $Z_1 = \operatorname{diag}(1, \zeta_2, \ldots, \zeta_m)$, *and* $D_2 = \{1 > \zeta_2 > \cdots > \zeta_m > 0\}$. *Then*

$$\int_{D_2} (\det Z)^{a-m} \prod_{k=2}^{m} (1 - \zeta_k)^2 \prod_{k<l}^{m} (\zeta_k - \zeta_l)^2 C_\kappa(Z_1) \bigwedge_{k=2}^{m} d\zeta_k = (ma + k) \frac{\mathcal{C}\Gamma_m(m)}{\pi^{m(m-1)}} \frac{\mathcal{C}\Gamma_m(a,\kappa)\mathcal{C}\Gamma_m(m)}{\mathcal{C}\Gamma_m(a+m,\kappa)} C_\kappa(I). \tag{4.2}$$

Proof. Let $b = m$ and $\zeta_k = \lambda_k/\lambda_1$, $k = 2, \ldots, m$. Then the left side of (4.1) becomes

$$\int_0^1 \lambda_1^{ma+k-1} d\lambda_1 \int_{D_2} (\det Z)^{a-m} \prod_{k=2}^{m} (1 - \zeta_k)^2 \prod_{k<l}^{m} (\zeta_k - \zeta_l)^2 C_\kappa(Z_1) \bigwedge_{k=2}^{m} d\zeta_k. \tag{4.3}$$

The result follows by noting that $\int_0^1 \lambda_1^{ma+k-1} d\lambda_1 = 1/(ma + k)$. □

The following theorem describes the joint density of the extreme eigenvalues of the central complex Wishart matrix.

THEOREM 4.3. *Let* $\mathbf{W} \sim \mathcal{C}W_m(n, \Sigma)$. *The joint distribution of* $\lambda_1 (= \lambda_{\max})$ *and*

$\lambda_m(=\lambda_{\min})$ of W is given by

$$f(\lambda_1,\lambda_m)=\frac{\pi^{m(m-1)}(\det\Sigma)^{-n}}{\mathcal{C}\Gamma_m(m)\mathcal{C}\Gamma_m(n)}\exp(-m\lambda_1)\sum_{k=0}^{\infty}\sum_{\kappa}\frac{\lambda_1^{mn+k-1}C_\kappa(\Sigma^{-1})}{k!\;C_\kappa(I)}\times\sum_{t=0}^{\infty}\sum_{\tau,\delta}\frac{(m-n)_\tau g_{\tau,\kappa}^{\delta}(1-\lambda_m/\lambda_1)^{(m-1)(m+1)+t+k-1}}{t!}\times[(m-1)(m+1)+k+t]\frac{\mathcal{C}\Gamma_{m-1}(m-1)}{\pi^{(m-1)(m-2)}}\times\frac{\mathcal{C}\Gamma_{m-1}(m+1,\delta)\mathcal{C}\Gamma_{m-1}(m-1)}{\mathcal{C}\Gamma_{m-1}(2m,\delta)}C_\delta(I),\tag{4.4}$$

where $g_{\tau,\kappa}^{\delta}$ is the coefficient of C_δ (defined in the proof).

Proof. Consider (3.2). By making the transformations $\lambda_1=\lambda_1$, $\eta_k=1-\lambda_k/\lambda_1$, $k=2,\dots,m$, we obtain the joint density of $\lambda_1,\eta_2,\dots,\eta_m$ as

$$\frac{\pi^{m(m-1)}(\det\Sigma)^{-n}}{\mathcal{C}\Gamma_m(m)\mathcal{C}\Gamma_m(n)}\exp(-m\lambda_1)\sum_{k=0}^{\infty}\sum_{\kappa}\lambda_1^{mn+k-1}(\det H)^2\det(I-H)^{n-m}\times\frac{C_\kappa(H)C_\kappa(\Sigma^{-1})}{k!\;C_\kappa(I)}\prod_{i>j=2}^{m}(\eta_i-\eta_j)^2,\quad 0<\lambda_1<\infty,\ 0<\eta_2<\cdots<\eta_m<1,$$

where $H=\operatorname{diag}(\eta_2,\dots,\eta_m)$. We have [9]

$$\begin{aligned}\det(I-H)^{n-m}C_\kappa(H)&=\sum_{t=0}^{\infty}\sum_{\tau}\frac{(-(n-m))_\tau C_\tau(H)C_\kappa(H)}{t!}\\&=\sum_{t=0}^{\infty}\sum_{\tau}\sum_{\delta}\frac{(-(n-m))_\tau g_{\tau,\kappa}^{\delta}C_\delta(H)}{t!},\end{aligned}$$

where $g_{\tau,\kappa}^{\delta}$ is the coefficient of $C_\delta(H)$ in the product $C_\tau(H)C_\kappa(H)$, $\delta=(\delta_1,\dots,\delta_m)$, $\delta_1\ge\cdots\ge\delta_m\ge 0$, and $\sum_{i=1}^{m}\delta_i=k+t$. Again, by making the transformations $\lambda_1=\lambda_1$, $\zeta_k=\eta_k/\eta_m$, $k=2,\dots,m-1$, and $\eta_m=\eta_m$, we obtain the joint density of $\lambda_1,\zeta_2,\dots,\zeta_{m-1}$, and η_m as

$$\frac{\pi^{m(m-1)}(\det\Sigma)^{-n}}{\mathcal{C}\Gamma_m(m)\mathcal{C}\Gamma_m(n)}\exp(-m\lambda_1)\sum_{k=0}^{\infty}\sum_{\kappa}\frac{\lambda_1^{mn+k-1}C_\kappa(\Sigma^{-1})}{k!\;C_\kappa(I)}\times\sum_{t=0}^{\infty}\sum_{\tau,\delta}\frac{(m-n)_\tau g_{\tau,\kappa}^{\delta}\eta_m^{(m-1)(m+1)+t+k-1}}{t!}\times(\det Z)^2C_\delta(Z_1)\prod_{i=2}^{m-1}(1-\zeta_i)^2\prod_{i>j=2}^{m-1}(\zeta_i-\zeta_j)^2,$$

where $Z=\operatorname{diag}(\zeta_2,\dots,\zeta_{m-1})$ and $Z_1=\operatorname{diag}(1,\zeta_2,\dots,\zeta_{m-1})$. Integrating with respect to $\zeta_2,\dots,\zeta_{m-1}$ and using Lemma 4.2, we obtain the joint density of λ_1 and η_m

as

(4.5)
$$g(\lambda_1,\eta_m)=\frac{\pi^{m(m-1)}(\det\Sigma)^{-n}}{\mathcal{C}\Gamma_m(m)\mathcal{C}\Gamma_m(n)}\exp(-m\lambda_1)\sum_{k=0}^{\infty}\sum_{\kappa}\frac{\lambda_1^{mn+k-1}C_\kappa(\Sigma^{-1})}{k!\;C_\kappa(I)}$$
$$\times\sum_{t=0}^{\infty}\sum_{\tau,\delta}\frac{(m-n)_\tau g^{\delta}_{\tau,\kappa}\eta_m^{(m-1)(m+1)+t+k-1}}{t!}[(m-1)(m+1)+k+t]$$
$$\times\frac{\mathcal{C}\Gamma_{m-1}(m-1)}{\pi^{(m-1)(m-2)}}\frac{\mathcal{C}\Gamma_{m-1}(m+1,\delta)\mathcal{C}\Gamma_{m-1}(m-1)}{\mathcal{C}\Gamma_{m-1}(2m,\delta)}C_\delta(I).$$

Finally, the result follows by substituting $\eta_m=1-\lambda_m/\lambda_1$. □

THEOREM 4.4. *Let* $\mathbf{W}=\mathbf{A}^H\mathbf{A}\sim\mathcal{C}W_m(n,\Sigma)$. *Since* $\operatorname{cond}(A)^2=\lambda_1/\lambda_m$, *then the density of* $y=1-1/\operatorname{cond}(A)^2$ *is given by*

(4.6)
$$f(y)=\frac{\pi^{m(m-1)}(\det\Sigma)^{-n}}{\mathcal{C}\Gamma_m(m)\mathcal{C}\Gamma_m(n)}\sum_{k=0}^{\infty}\sum_{\kappa}\frac{\Gamma(mn+k)C_\kappa(\Sigma^{-1})}{m^{mn+k}\;k!\;C_\kappa(I)}$$
$$\times\sum_{t=0}^{\infty}\sum_{\tau,\delta}\frac{(m-n)_\tau g^{\delta}_{\tau,\kappa}y^{(m-1)(m+1)+t+k-1}}{t!}[(m-1)(m+1)+k+t]$$
$$\times\frac{\mathcal{C}\Gamma_{m-1}(m-1)}{\pi^{(m-1)(m-2)}}\frac{\mathcal{C}\Gamma_{m-1}(m+1,\delta)\mathcal{C}\Gamma_{m-1}(m-1)}{\mathcal{C}\Gamma_{m-1}(2m,\delta)}C_\delta(I).$$

Proof. The result follows by integrating (4.5) with respect to λ_1 and substituting $y=\eta_m$. Note that we have

$$\int_0^\infty e^{-m\lambda_1}\lambda_1^{mn+k-1}\,d\lambda_1=\frac{\Gamma(mn+k)}{m^{mn+k}}.$$ □

If $\Sigma=\sigma^2 I$, then the corresponding results for Theorems 4.3 and 4.4 can be derived using a similar method. However, we provide an alternative approach as follows.

THEOREM 4.5. *Let* $\Sigma=\sigma^2 I$. *The joint density of* $\lambda_1(=\lambda_{\max})$ *and* $\lambda_m(=\lambda_{\min})$ *of a central Wishart matrix is given by*

(4.7)
$$f(\lambda_1,\lambda_m)=\frac{\pi^{m(m-1)}(\sigma^2)^{-nm}}{\mathcal{C}\Gamma_m(m)\mathcal{C}\Gamma_m(n)}\lambda_1^{(m-1)(n-m-1)+m}\exp\left\{-\frac{1}{\sigma^2}[(m-1)\lambda_1-\lambda_m]\right\}\lambda_m^{n-m}$$
$$\times(\lambda_1-\lambda_m)^{m^2-2}\varrho(\psi;m-2,2,0,1),\quad 0<\lambda_m<\lambda_1<\infty,$$

where

(4.8)
$$\varrho(\psi,m,r,L,U)=\int_{D_3}\prod_{k=1}^{m}(x_k^r\psi(x_k))\prod_{k>l=1}^{m}(x_k-x_l)^2\bigwedge_{k=1}^{m}dx_k,$$

and $\psi(x)=(1-x)^2(1-x-(\lambda_m/\lambda_1)x)^{n-m}\exp\left(\frac{1}{\sigma^2}(\lambda_1-\lambda_m)x\right)$ *with* $D_3=\{L\le x_1\le\cdots\le x_q\le U\}$.

Proof. Consider (3.3). By making the transformations $\lambda_1=\lambda_1$, $\eta_k=1-\lambda_k/\lambda_1$,

$k = 2, \ldots, m$, we obtain the joint density of $\lambda_1, \eta_2, \ldots, \eta_m$ as

$$\frac{\pi^{m(m-1)}(\sigma^2)^{-nm}}{\mathcal{C}\Gamma_m(m)\mathcal{C}\Gamma_m(n)}\lambda_1^{mn-1}\exp\left(-\frac{1}{\sigma^2}m\lambda_1\right) \times \prod_{k=2}^{m}\left[\eta_k^2(1-\eta_k)^{n-m}\exp\left(\frac{1}{\sigma^2}\lambda_1\eta_k\right)\right]\prod_{k>l=2}^{m}(\eta_k-\eta_l)^2,$$

where $0 < \lambda_1 < \infty$ and $0 < \eta_2 < \cdots < \eta_m < 1$. Again, by making the transformations $\lambda_1 = \lambda_1$, $\zeta_k = \eta_k/\eta_m$, $k = 2, \ldots, m-1$, and $\eta_m = \eta_m$, we obtain the joint density of $\lambda_1, \zeta_2, \ldots, \zeta_{m-1}$, and η_m as

$$\frac{\pi^{m(m-1)}(\sigma^2)^{-nm}}{\mathcal{C}\Gamma_m(m)\mathcal{C}\Gamma_m(n)}\lambda_1^{mn-1}\exp\left\{-\frac{1}{\sigma^2}\lambda_1(m-\eta_m)\right\}\eta_m^{m^2-2}(1-\eta_m)^{n-m} \times \prod_{k=2}^{m-1}\left[\zeta_k^2(1-\zeta_k)^2(1-\eta_m\zeta_k)^{n-m}\exp\left(\frac{1}{\sigma^2}\lambda_1\eta_m\zeta_k\right)\right]\prod_{k>l=2}^{m-1}(\zeta_k-\zeta_l)^2,$$

where $0 < \lambda_1 < \infty$, $0 < \zeta_2 < \cdots < \zeta_{m-1} < 1$, and $0 < \eta_m < 1$. Upon integration with respect to $\zeta_2, \ldots, \zeta_{m-1}$, the joint density, $g(\lambda_1, \eta_m)$, of λ_1 and η_m is given by

$$\frac{\pi^{m(m-1)}(\sigma^2)^{-nm}}{\mathcal{C}\Gamma_m(m)\mathcal{C}\Gamma_m(n)}\lambda_1^{mn-1}\exp\left\{-\frac{1}{\sigma^2}\lambda_1(m-\eta_m)\right\}\eta_m^{m^2-2} \times (1-\eta_m)^{n-m}\varrho(\psi; m-2, 2, 0, 1), \tag{4.9}$$

where

$$\psi(x) = (1-x)^2(1-\eta_m x)^{n-m}\exp\left(\frac{1}{\sigma^2}\lambda_1\eta_m x\right), \quad 0 < \lambda_1 < \infty,$$

and $0 < \eta_m < 1$. Now, the result follows by substituting $\eta_m = 1 - \lambda_m/\lambda_1$. □

THEOREM 4.6. *Let* $\mathbf{W} = \mathbf{A}^H\mathbf{A} \sim \mathcal{C}W_m(n, \sigma^2 I)$. *Since* $\operatorname{cond}(A)^2 = \lambda_1/\lambda_m$, *then the density of* $y = 1 - 1/\operatorname{cond}(A)^2$ *is given by*

$$f(y) = \int_0^\infty g(\lambda_1, \eta_m)\, d\lambda_1, \quad 0 < y < 1. \tag{4.10}$$

Proof. The proof is obvious from (4.9). □

It should be noted that the joint density of the extreme eigenvalues of a real central Wishart matrix is studied in [16], [17], [20], and [21]. The density given in Theorem 4.6 may be used to test the sphericity hypothesis $H_0 : \Sigma = \sigma^2 I$ against the alternative $H_1 : \Sigma \neq \sigma^2 I$; see [17]. It may also be used to test the sphericity hypothesis against the alternative that any two eigenvalues of Σ are unequal.

Consider the following numerical example for testing the sphericity hypothesis. A sequence of 23 complex signals is received at the output of the communication system. Let the number of outputs of the system be 3 ($m = 3$). The sample covariance matrix is given by

$$S = \begin{bmatrix} 150.77 & 78.15 + 15.12i & 35.32 + 10.15i \\ 78.15 - 15.12i & 71.05 & 23.65 + 10.12i \\ 35.32 - 10.15i & 23.65 - 10.12i & 12.26 \end{bmatrix}.$$

Assume the complex multivariate signal is normal with population covariance matrix Σ. Then $W = 22S$ is a complex central Wishart distribution with 22 degrees of freedom, $W \sim \mathcal{C}W_3(22, \Sigma)$. We wish to test the sphericity hypothesis $H_0 : \Sigma = \sigma^2 I$, σ^2 unknown against H_1: any two eigenvalues of Σ are unequal at the significance level $\alpha = 0.05$. Let $\lambda_1 > \lambda_2 > \lambda_3$ be the eigenvalues of W. Then the critical region is given by

$$\frac{\lambda_1}{\lambda_3} = \text{cond}(W) \geq c,$$

where the constant c is chosen to make the significance level equal to 0.05. This critical region can be written equivalently as

$$y \geq d,$$

where $y = 1 - \lambda_3/\lambda_1$, the density $f(y)$ is given in (4.10), and d is a constant chosen to make the significance level equal to 0.05. Thus d is chosen such that

$$P_{H_0}(y \geq d) = \int_d^1 f(y)\, dy = 0.05.$$

A numerical evaluation of this probability shows that $d = 0.7$ with $m = 3,\ n = 22$, and $\sigma^2 = 1$. For the measured data we have $y = 1 - \lambda_3/\lambda_1 = 1 - 2/209.88 = 0.9905$, which is highly significant at the 5% level, and so we reject the sphericity hypothesis. If $W \sim \mathcal{C}W_3(22, I_3)$, it also follows from this calculation that $P(\text{cond}(W) > 1/(1-d)) = 0.05$.

5. Conclusion. In this paper, the distributions of the largest and smallest eigenvalues of a complex Wishart matrix were derived for an arbitrary covariance matrix Σ, and the joint distributions of the extreme eigenvalues were also derived. Using these distributions we derived the condition number distributions of complex random matrices. These distributions play an important role in numerical analysis and statistical hypothesis testing.

REFERENCES

[1] T. Baker and P. Forrester, *The Calogero-Sutherland model and generalized classical polynomials*, Comm. Math. Phys., 188 (1997), pp. 175–216.

[2] A. G. Constantine, *Some noncentral distribution problems in multivariate analysis*, Ann. Math. Statist., 34 (1963), pp. 1270–1285.

[3] A. Edelman, *Eigenvalues and condition numbers of random matrices*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 543–560.

[4] H. H. Goldstine and J. von Neumann, *Numerical inverting of matrices of high order* II, Amer. Math. Soc. Proc., 2 (1951), pp. 188–202.

[5] A. T. James, *Distributions of matrix variate and latent roots derived from normal samples*, Ann. Math. Statist., 35 (1964), pp. 475–501.

[6] I. M. Johnstone, *On the distribution of the largest eigenvalue in principal components analysis*, Ann. Statist., 29 (2001), pp. 295–327.

[7] C. G. Khatri, *Distribution of the largest or the smallest characteristic root under null hypothesis concerning complex multivariate normal populations*, Ann. Math. Statist., 35 (1964), pp. 1807–1810.

[8] C. G. Khatri, *On certain distribution problems based on positive definite quadratic functions in normal vectors*, Ann. Math. Statist., 37 (1966) pp. 468–479.

[9] C. G. Khatri and K. C. S. Pillai, *On the non-central distributions of two test criteria in multivariate analysis of variance*, Ann. Math. Statist., 39 (1968), pp. 215–226.

[10] I. G. Macdonald, *Symmetric Functions and Hall Polynomials*, Oxford University Press, New York, 1995.

[11] M. L Mehta, *Random Matrices*, 2nd ed., Academic Press, New York, 1991.
[12] R. J. Muirhead, *Aspects of Multivariate Statistical Theory*, Wiley, New York, 1982.
[13] J. Shen, *On the singular values of Gaussian random matrices*, Linear Algebra Appl., 326 (2001), pp. 1–14.
[14] S. Smale, *On the efficiency of algorithms of analysis*, Bull. Amer. Math. Soc., 13 (1985), pp. 87–121.
[15] R. P. Stanley, *Some combinatorial properties of Jack symmetric functions*, Adv. Math., 77, (1989), pp. 76–115.
[16] T. Sugiyama, *On the distribution of the largest latent root of the covariance matrix*, Ann. Math. Statist., 38 (1967), pp. 1148–1151.
[17] T. Sugiyama, *Joint distribution of the extreme roots of a covariance matrix*, Ann. Math. Statist., 41 (1970), pp. 655–657.
[18] T. Sugiyama, *Distributions of the largest latent root of the multivariate complex Gaussian distribution*, Ann. Inst. Statist. Math., 24 (1972), pp. 87–94.
[19] J. von Neumann and H. H. Goldstine, *Numerical inverting of matrices of high order*, Bull. Amer. Math. Soc., 53 (1947), pp. 1021–1099.
[20] V. B. Waikar and F. J. Schuurmann, *Exact joint density of the largest and smallest roots of the Wishart and MANOVA matrices*, Util. Math., 4 (1973), pp. 253–260.
[21] V. B. Waikar, *On the joint distributions of the largest and smallest latent roots of two random matrices (noncentral case)*, South African Statist. J., 7 (1973), pp. 103–108.
[22] J. H. Wilkinson, *Error analysis revisited*, IMA Bulletin, 22 (1986), pp. 192–200.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 2, pp. 457–476

EFFICIENT ALGORITHMS FOR SOLUTION OF REGULARIZED TOTAL LEAST SQUARES*

ROSEMARY A. RENAUT† AND HONGBIN GUO†

Abstract. Error-contaminated systems $Ax \approx b$, for which A is ill-conditioned, are considered. Such systems may be solved using Tikhonov-like regularized total least squares (RTLS) methods. Golub, Hansen, and O'Leary [*SIAM J. Matrix Anal. Appl.*, 21 (1999), pp. 185–194] presented a parameter-dependent direct algorithm for the solution of the augmented Lagrange formulation for the RTLS problem, and Sima, Van Huffel, and Golub [*Regularized Total Least Squares Based on Quadratic Eigenvalue Problem Solvers*, Tech. Report SCCM-03-03, SCCM, Stanford University, Stanford, CA, 2003] have introduced a technique for solution based on a quadratic eigenvalue problem, RTLSQEP. Guo and Renaut [*A regularized total least squares algorithm*, in Total Least Squares and Errors-in-Variables Modeling: Analysis, Algorithms and Applications, S. Van Huffel and P. Lemmerling, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002, pp. 57–66] derived an eigenproblem for the RTLS which can be solved using the iterative inverse power method. Here we present an alternative derivation of the eigenproblem for constrained TLS through the augmented Lagrangian for the constrained normalized residual. This extends the analysis of the eigenproblem and leads to derivation of more efficient algorithms compared to the original formulation. Additional algorithms based on bisection search and a standard L-curve approach are presented. These algorithms vary with respect to the parameters that need to be prescribed. Numerical and convergence results supporting the different versions and contrasting with RTLSQEP are presented.

Key words. total least squares, regularization, ill-posedness, Rayleigh quotient iteration

AMS subject classifications. 65F22, 65F30

DOI. 10.1137/S0895479802419889

1. Introduction.

We consider the solution of the ill-posed model dependent problem

$$Ax \approx b,$$

where $A \in R^{m\times n}$ and $b \in R^m$ are known, and are assumed to be error contaminated. If the matrix A is well conditioned a solution can be found using the method of total least squares (TLS)

$$\min\|[E, f]\|_F \quad \text{subject to} \quad (A+E)x = b+f, \tag{1.1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm [4, 5, 12].

For ill-conditioned systems, Golub, Hansen, and O'Leary [3] presented and analyzed the properties of regularization for TLS. Consistent with the formulation of the Tikhonov regularized LS problem [15, 16], the regularized TLS (RTLS) is given by

$$\min\|[E, f]\|_F \quad \text{subject to} \quad (A+E)x = b+f \quad \text{and} \quad \|Lx\| \le \delta. \tag{1.2}$$

*Received by the editors December 13, 2002; accepted for publication (in revised form) by P. C. Hansen January 29, 2004; published electronically January 12, 2005. This research was supported in part by the Arizona Alzheimer's Disease Research Center, which is funded by the Arizona Department of Health Services, and NIH grant EB 2553301. The first author is also supported by NSF grant CMG-02223 and acknowledges the support of the Technical University of Munich through the award of the John von Neumann visiting professorship in 2001–2002.

http://www.siam.org/journals/simax/26-2/41988.html

†Department of Mathematics and Statistics, Arizona State University, Tempe, AZ 85287-1804 (renaut@asu.edu, hb_guo@asu.edu).

Here $\|\cdot\|$ denotes the 2-norm, δ is a regularization parameter, and $L \in R^{p\times n}$ defines a (semi)norm on the solution [9, 3].

Guo and Renaut [6] obtained the solution of the RTLS problem by finding the minimum eigenpair for an augmented solution-dependent block matrix. The eigenpair is found iteratively, using inverse iteration applied to the solution-dependent matrix. Here we present a theoretical development of a convergent algorithm for determination of the minimum eigenpair. The algorithm is extended to improve efficiency by inclusion of the option to do inexact local solves and update of the constraint within the matrix formulation. Bisection search is presented because of the ability to determine precisely the number of iterations required for a given accuracy. We also provide an L-curve approach for cases in which a good estimate of the physical constraint parameter is not available.

Theoretical results which lead to the development of the algorithms are presented in section 2. The theory uses the alternative statement of the RTLS problem, namely the minimization of the normalized residual, equivalently the minimization of the Rayleigh quotient for the augmented matrix

$$M = [A, b]^T[A, b] \tag{1.3}$$

[12] subject to the addition of the constraint term for regularization of the solution. We also present results on the relationships between the Lagrange multipliers of the RTLS and the constraint parameter δ. If any one of the set of three parameters is chosen as a free parameter, the other two are immediately specified and are monotonically related to one another. This result verifies the connection between the presented algorithms. Computational details are described in section 3, and experimental results comparing and contrasting the different approaches and comparing with the quadratic eigenvalue solver presented in [14] are described in section 4. We conclude that the eigenproblem formulation provides a powerful approach for RTLS solutions in practical applications.

2. Algorithmic development.

2.1. Rayleigh quotient formulation. It is well known that the solution of the TLS minimizes the sum of squared normalized residuals,

$$x_{TLS} = \text{argmin}_x \phi(x) = \text{argmin}_x \frac{\|Ax - b\|^2}{1 + \|x\|^2} \tag{2.1}$$

[5, 12], where ϕ is the Rayleigh quotient of matrix M. This suggests an alternative formulation for regularized TLS,

$$\min_x \phi(x) \;\text{ subject to } \|Lx\| \leq \delta. \tag{2.2}$$

To distinguish the two formulations we call this the regularized Rayleigh quotient form for total least squares (RQ-RTLS). It leads to the augmented Lagrangian

$$\mathcal{L}(x, \mu) = \phi(x) + \mu(\|Lx\|^2 - \delta^2). \tag{2.3}$$

Although $\phi(x)$ is not concave its stationary points can be characterized, which is useful in characterization of the solution of (2.3).

LEMMA 2.1 (Fact 1.8 in [13]). *The Rayleigh quotient of a symmetric matrix is stationary at, and only at, the eigenvectors of the matrix.*

LEMMA 2.2. *If the extreme singular values of the matrix* $[A, b]$ *are simple, then* $\phi(x)$ *has one unique maximum point, one unique minimum point, and* $n-1$ *saddle points.*

Proof. The proof follows by the observation $\phi(x_i) = \sigma_i^2$, where vector $[x_i^T, -1]^T$ is the ith right singular vector of matrix $[A, b]$ with corresponding singular value σ_i. The uniqueness of the maximum and minimum points is immediate. To show that all the other stationary points are saddles, it is easy to construct their neighbors and show that $\phi(x)$ is, resp., greater and less on either side of the corresponding stationary point. ∎

THEOREM 2.1. *Suppose that the conditions of Lemma* 2.2 *are satisfied for matrix* $[A, b]$*, that* $\sigma_n > \sigma_{n+1}$ *and that constraint parameter* δ *is specified. Then, if the constraint is active,* $\|Lx_{RTLS}\|^2 = \delta^2$ *and* $\mu > 0$.

Proof. By Lemma 2.2 and using $\sigma_n > \sigma_{n+1}$, the solution x_{TLS} of (2.1) is unique. If the constraint in (2.2) is active, $\|Lx_{TLS}\|^2 > \delta^2$, and by Lemma 2.2 there is no local minimum of ϕ within the set defined by the constraint $\|Lx\|^2 < \delta^2$. Thus, if the constraint is active, x_{RTLS} must lie on the boundary of the domain defined by $\|Lx\|^2 \leq \delta^2$,

$$x_{RTLS}^T L^T L x_{RTLS} - \delta^2 = 0, \tag{2.4}$$

and the Lagrange parameter at the minimum of the Lagrangian is positive, $\mu > 0$. ∎

It is easy to see that the Kuhn–Tucker conditions for (2.2) are the same as those for (1.2). Hence we immediately obtain the following theorem for the characterization of the RTLS solution for (2.2), equivalent to that presented in [3] for the augmented Lagrangian for (1.2).

THEOREM 2.2. *The solution,* x_{RTLS}*, of the regularized problem* (2.2)*, for which the constraint is active, satisfies*

$$(A^T A + \lambda_I I + \lambda_L L^T L) x_{RTLS} = A^T b, \tag{2.5}$$

$$\mu > 0, \quad x_{RTLS}^T L^T L x_{RTLS} - \delta^2 = 0, \tag{2.6}$$

where

$$\lambda_I = -\phi(x_{RTLS}), \tag{2.7}$$

$$\lambda_L = \mu(1 + \|x_{RTLS}\|^2), \tag{2.8}$$

$$\mu = -\frac{1}{\delta^2(1 + \|x_{RTLS}\|^2)} \left(b^T (A x_{RTLS} - b) + \phi(x_{RTLS}) \right). \tag{2.9}$$

Proof. Setting $\nabla_x \mathcal{L}(x, \mu) = 0$, we have

$$A^T A x - A^T b + \mu(1 + \|x\|^2) L^T L x - \phi(x) x = 0,$$

which is (2.5) with λ_I and λ_L identified by (2.7) and (2.8), resp. Multiplying both sides by x^T, replacing $\|Lx\|$ by δ, and using the relationship (2.1) to rewrite $\|Ax\|^2 - b^T A x$ as $(1+\|x\|^2)\phi(x) + b^T(Ax - b)$, we immediately obtain the expression for μ. Moreover, (2.6) follows from Theorem 2.1. ∎

In [6] we observed, without proof, that this result additionally characterizes the RTLS solution in terms of an eigenpair for an augmented system. Here we present a slight modification of the result, of significant practical use, which includes the constraint condition in an alternative manner.

THEOREM 2.3. *The solution x_{RTLS} of (2.2) subject to the active constraint (2.4) satisfies the augmented eigenpair problem:*

$$B(x_{RTLS})\begin{pmatrix} x_{RTLS} \\ -1 \end{pmatrix} = -\lambda_I \begin{pmatrix} x_{RTLS} \\ -1 \end{pmatrix}, \tag{2.10}$$

where the solution-dependent matrix is given by

$$B(x_{RTLS}) = M + \lambda_L(x_{RTLS}) \begin{pmatrix} L^T L & 0 \\ 0 & \alpha \end{pmatrix}, \qquad \alpha = -\|Lx_{RTLS}\|^2, \tag{2.11}$$

in which $\lambda_L(x_{RTLS})$ is given by (2.8) and (2.9).

Conversely, suppose the pair $((\hat{x}^T, -1)^T, -\hat{\lambda})$ is an eigenpair for matrix $\hat{B}(\hat{x})$, where matrix $\hat{B}$ represents the matrix B with modification in the lower right corner, α replaced by $\hat{\alpha}$, $\hat{\alpha} = -\gamma$, where γ can take values δ^2, or $\|L\hat{x}\|^2$, and $\lambda_L(\hat{x})$ is defined accordingly by (2.8) and (2.9). Then

1. *$\hat{x}$ satisfies (2.5),*
2. *the constraint is active, (2.4) is satisfied for $\hat{x}$, and*
3. *eigenvalue $\hat{\lambda}$ is given by*

$$\hat{\lambda} = -\phi(\hat{x}).$$

Proof. The first block equation of (2.10) comes immediately from (2.5). For the second block equation we note that by (2.7)

$$\lambda_I(1 + \|x\|^2) = -\|Ax - b\|^2,$$

but by (2.5)

$$\lambda_I\|x\|^2 = b^T Ax - \|Ax\|^2 - \lambda_L\|Lx\|^2.$$

Thus, by subtraction,

$$\lambda_I = b^T Ax + \lambda_L x\|Lx\|^2 - b^T b, \tag{2.12}$$

as required. We replace $\|Lx\|^2$ occurring in α by δ^2 using the active constraint condition (2.4).

For the proof in the opposite direction, we suppose that the eigenpair $((\hat{x}^T, -1)^T, -\hat{\lambda})$ satisfies the eigenvalue equation (2.10), with appropriate replacement of x_{RTLS} by $\hat{x}$ and λ_I by $\hat{\lambda}$. The first block equation immediately gives (2.5). By the second block equation of the eigenvalue problem we have

$$\hat{\lambda} = b^T A\hat{x} - b^T b + \lambda_L(\hat{x})\gamma, \tag{2.13}$$

and by the inner product of the eigensystem equation with eigenvector $(\hat{x}^T, -1)^T$ we have

$$\hat{\lambda} = -\frac{1}{\|\hat{x}\|^2 + 1}(\|A\hat{x} - b\|^2 + \lambda_L(\hat{x})(\|L\hat{x}\|^2 - \gamma)). \tag{2.14}$$

Equating these two expressions, collecting terms in λ_L and then using (2.8) and (2.9) we find

$$\lambda_L\left(\frac{\|L\hat{x}\|^2 - \gamma}{\|\hat{x}\|^2 + 1} + \gamma\right) = \lambda_L\delta^2.$$

Solving for γ, by using the fact that $\lambda_L \neq 0$, because it is proportional to $\mu \neq 0$, yields

$$\gamma = \frac{\delta^2(1 + \|\hat{x}\|^2) - \|L\hat{x}\|^2}{\|\hat{x}\|^2}, \tag{2.15}$$

which is satisfied for $\gamma = \delta^2$, or $\gamma = \|L\hat{x}\|^2$, each of which also imposes the active constraint equation (2.4). When inserted back into the second expression for $\hat{\lambda}$ this also yields

$$\hat{\lambda} = -\frac{\|A\hat{x} - b\|^2}{\|\hat{x}\|^2 + 1} - \lambda_L(\hat{x})\frac{\|L\hat{x}\|^2 - \delta^2}{\|\hat{x}\|^2},$$

where now the second term vanishes because both of the choices for γ also enforce the active constraint condition. Hence $\hat{\lambda}(\hat{x}) = \lambda_I(\hat{x})$, as required. ∎

2.2. Theoretical development. By the definition of RTLS problem (2.2) and Theorem 2.3, the RTLS solution can be obtained through estimation of the minimum $|\lambda_I| = \phi(x)$ which solves the augmented eigenvalue problem. Whenever the system (2.10) is satisfied, the active constraint condition is also immediately satisfied. To derive practical algorithms for the solution of the eigenproblem, we observe the similarity with the unconstrained TLS problem: $(x_{TLS}^T, -1)^T$ is a right eigenvector for matrix M associated with its smallest eigenvalue. An algorithm based on the Rayleigh quotient iteration (RQI) for matrix (1.3) was presented by Björck, Heggernes, and Matstoms [2]. While a similar iterative approach can be implemented, there is the additional complication that the system matrix (2.11) depends on the solution x_{RTLS}, which requires consideration of the convergence properties applied to this particular situation. On the other hand, in [6], we verified numerically that inverse iteration can be used for the determination of the RTLS solution. Here we investigate the convergence properties of the approach and introduce modifications of the algorithm to improve efficiency and reliability.

To analyze the eigenproblem for (2.10), for the case in which we use $\gamma = \delta^2$, we introduce the parameter-dependent matrix, $\mathbf{B}(\theta) = M + \theta N$, $\theta \in R^+$, where

$$N = \begin{pmatrix} L^T L & 0 \\ 0 & -\delta^2 \end{pmatrix}. \tag{2.16}$$

Obviously $\mathbf{B}(\lambda_L) = B(x)$ in (2.11) if λ_L is given by (2.8) with x in place of x_{RTLS}. We denote the smallest eigenvalue corresponding to eigenvector $(x_\theta^T, -1)^T$ of $\mathbf{B}(\theta)$ by ϱ_{n+1}, and use the notation $\mathcal{N}(A)$ for the null space of matrix A.

We also introduce the function $g(x) = (\|Lx\|^2 - \delta^2)/(1 + \|x\|^2)$. Then the goal of solving the augmented eigenproblem may be reformulated as follows.

PROBLEM 2.4. *For a constant δ, find a θ such that $g(x_\theta) = 0$.*

The following results assist with the design of an algorithm to solve this problem.

LEMMA 2.3. *Assuming that $b^T A \neq 0$ and $\mathcal{N}(A) \cap \mathcal{N}(L) = \{0\}$, then the smallest eigenvalue of $\boldsymbol{B}(\theta)$ is simple.*

Proof. The eigenvalue-eigenvector equation

$$\mathbf{B}(\theta)\begin{pmatrix} x_\theta \\ -1 \end{pmatrix} = \rho_\theta \begin{pmatrix} x_\theta \\ -1 \end{pmatrix}$$

yields

$$(A^T A + \theta L^T L - \rho_\theta I)x_\theta = A^T b. \tag{2.17}$$

By assumption $A^T b \neq 0$, so ρ_θ is not an eigenvalue of $A^T A + \theta L^T L$. By the eigenvalue interlace theorem, it is thus strictly smaller than the smallest eigenvalue of $A^T A + \theta L^T L$ and must be simple. ∎

LEMMA 2.4. *If $[A, b]$ is a full rank matrix, there exists one and only one positive number, denoted by θ^c, such that $\boldsymbol{B}(\theta^c)$ is singular, and*

1. *the null eigenvalue of $\boldsymbol{B}(\theta^c)$ is simple,*
2. *when $0 \leq \theta < \theta^c$, $\boldsymbol{B}(\theta)$ is positive definite, and*
3. *when $\theta > \theta^c$, $\boldsymbol{B}(\theta)$ has only one negative eigenvalue; the others are positive.*

Proof. Because M is nonsingular, $\mathbf{B}(\theta) = M + \theta N$ is congruent to $C(\theta) = I + \theta X^T N X$, where X is a nonsingular matrix. Thus $\mathbf{B}(\theta)$ and $C(\theta)$ have the same inertia, as do N and $X^T N X$. Because $L^T L$ is nonnegative definite, we know $C(\theta)$ is similar to

$$I + \theta \begin{pmatrix} D & & \\ & 0 & \\ & & -\omega^2 \end{pmatrix}, \tag{2.18}$$

where D is a diagonal matrix with positive diagonal entries. Thus there exists only one finite real number $\theta = \theta^c > 0$ such that the null space of $\mathbf{B}(\theta)$ is nontrivial and the dimension of the corresponding null space is 1.

Because matrices (2.18) and $\mathbf{B}(\theta)$ have the same inertia, we immediately obtain the other two results. ∎

LEMMA 2.5. *If $b^T A \neq 0$, and $[A, b]$ is full rank, then*

1. *there exists a $\lambda_L^* \in [0, \theta^c]$ which solves Problem 2.4,*
2. *the solution of Problem 2.4 is unique,*
3. *when $\lambda_L \in (0, \lambda_L^*)$, $g(x_{\lambda_L}) > 0$ and $\lambda_L \in (\lambda_L^*, \infty)$, $g(x_{\lambda_L}) < 0$.*

Proof.

1. When $\theta = 0$, $\mathbf{B}(0) > 0$. The eigenvector corresponding to the smallest eigenvalue of $\mathbf{B}(0)$, (M), is related to the TLS solution x_{TLS}, $g(x_{TLS}) > 0$ because the constraint is active. Moreover, for small perturbation in the matrix $\mathbf{B}(\theta)$, Theorem 6.3.12 in [11] yields

$$\frac{d\varrho_{n+1}}{d\theta}|_{\theta=\theta_0} = g(x_{\theta_0}). \tag{2.19}$$

   Thus, ϱ_{n+1} increases with θ near zero. On the other hand, by Lemma 2.4 $\varrho_{n+1} = 0$ for $\theta = \theta^c$. Thus, $g(x_\theta)$ must change sign in $[0, \theta^c]$ and by continuity there must exist a number $\lambda_L^* \in [0, \theta^c]$ such that the corresponding $g(x_{\lambda_L^*}) = 0$. Hence Problem 2.4 is solved.
2. We introduce notation x_θ,

$$x_\theta = \text{argmin}_{x \in R^n}(\phi(x) + \theta g(x)). \tag{2.20}$$

   Clearly, by Lemma 2.3, the smallest eigenvalue of $\mathbf{B}(\theta)$ is simple. Suppose that vectors $x_{\theta_1}, x_{\theta_2}$ solve (2.20) for $\theta_1, \theta_2 > 0$; then

$$\phi(x_{\theta_2}) + \theta_2 g(x_{\theta_2}) < \phi(x_{\theta_1}) + \theta_2 g(x_{\theta_1}), \tag{2.21}$$

$$\phi(x_{\theta_1}) + \theta_1 g(x_{\theta_1}) < \phi(x_{\theta_2}) + \theta_1 g(x_{\theta_2}). \tag{2.22}$$

   Adding these inequalities yields

$$(\theta_1 - \theta_2) g(x_{\theta_1}) < (\theta_1 - \theta_2) g(x_{\theta_2}), \tag{2.23}$$

and without loss of generality assuming $\theta_1 > \theta_2$, $g(x_{\theta_1}) < g(x_{\theta_2})$. Thus, $g(x_\theta)$ is monotonically decreasing with respect to θ and there exists only one θ such that $g(x_\theta) = 0$.

3. The final statement follows immediately from the former two. ∎

REMARK 2.1. *We see from this result that there is a unique solution to our problem and that an algorithm for finding this solution should depend both on finding an update for the Lagrange parameter λ_L and monitoring the sign of $g(x_{\lambda_L})$.*

From (2.13) it is immediate that x_θ is related to θ by $\theta = \frac{1}{\delta^2}(b^T(b - Ax_\theta) - \phi(x_\theta))$. This suggests an iterative search for the θ,

$$\theta^{(k+1)} = \frac{1}{\delta^2}(b^T(b - Ax_{\theta^{(k)}}) - \phi(x_{\theta^{(k)}})), \tag{2.24}$$

where, at step k, $(x_{\theta^{(k)}}^T, -1)^T$ is the eigenvector for $\varrho_{n+1}^{(k)}$. On the other hand, by (2.14), we can write $\varrho_{n+1}^{(k)} = \phi(x_{\theta^{(k)}}) + \theta^{(k)} g(x_{\theta^{(k)}})$, which in (2.24), also using $b^T A x_{\theta^{(k)}} - b^T b + \delta^2 \theta^{(k)} = -\varrho_{n+1}^{(k)}$, gives an update equation

$$\theta^{(k+1)} = \theta^{(k)} + \frac{\theta^{(k)}}{\delta^2} g(x_{\theta^{(k)}}). \tag{2.25}$$

It remains to consider whether this iteration will generate the appropriate θ that solves Problem 2.4. We investigate the convergence properties of the update equation (2.25), but first revert to the use of the parameter λ_L in place of θ. The theory presented in Lemma 2.5 suggests the use of an iteration dependent parameter $0 < \iota^{(k)} \leq 1$ chosen such that $g(x_{\lambda_L^{(k+1)}})$ has the same sign as $g(x_{\lambda_L^{(0)}})$:

$$\lambda_L^{(k+1)} = \lambda_L^{(k)} + \iota^{(k)} \frac{\lambda_L^{(k)}}{\delta^2} g(x_{\lambda_L^{(k)}}), \qquad 0 < \iota^{(k)} \leq 1. \tag{2.26}$$

LEMMA 2.6. *Suppose $\lambda_L^{(0)} > 0$. Let sequences $\{x_{\lambda_L^{(k)}}\}$ and $\{\lambda_L^{(k)}\}$, $k = 1, 2, \ldots$, be generated by (2.26), with parameter sequence $0 < \iota^{(k)} \leq 1$ utilized to enforce $g(x_{\lambda_L^{(k+1)}}) g(x_{\lambda_L^{(0)}}) > 0$.*

1. *$\lambda_L^{(k)} > 0$ for any positive integer k.*
2. *If $g(x_{\lambda_L^{(0)}}) < 0$, then sequences $\{\lambda_L^{(k)}\}$ and $\{\phi(x_{\lambda_L^{(k)}})\}$ decrease monotonically, while $\{\varrho_{n+1}^{(k)}\}$ and $\{g(x_{\lambda_L^{(k)}})\}$ increase monotonically.*
3. *If $g(x_{\lambda_L^{(0)}}) > 0$, then sequences $\{\lambda_L^{(k)}\}$ and $\{\phi(x_{\lambda_L^{(k)}})\}$ increase monotonically, while $\{\varrho_{n+1}^{(k)}\}$ and $\{g(x_{\lambda_L^{(k)}})\}$ decrease monotonically.*
4. *If $g(x_{\lambda_L^{(0)}}) = 0$, $\lambda_L^{(0)}$ solves Problem 2.4.*

Proof. For ease of presentation we write $x^{(k)}$ to indicate $x_{\lambda_L^{(k)}}$, assuming the dependence of the update on the $\lambda_L^{(k)}$.

1. Multiplying both sides of (2.26) by $\delta^2(1 + \|x^{(k)}\|^2)$, we obtain

$$\delta^2(1 + \|x^{(k)}\|^2)\lambda_L^{(k+1)} = \delta^2(1 - \iota^{(k)} + \|x^{(k)}\|^2)\lambda_L^{(k)} + \iota^{(k)}\lambda_L^{(k)}\|Lx^{(k)}\|^2.$$

Because $\iota^{(k)} \leq 1$, $\lambda_L^{(k+1)} > 0$ if $\lambda_L^{(k)} > 0$. Thus $\lambda_L^{(0)} > 0$ ensures $\lambda_L^{(k)} > 0$ for all $k \geq 1$.

2. If $g(x^{(0)}) < 0$, the algorithm forces $g(x^{(k)}) < 0$ for $k > 1$ so that also $\lambda_L^{(k+1)} < \lambda_L^{(k)}$. Then by (2.23) $g(x^{(k)}) < g(x^{(k+1)}) < 0$ and combining with (2.21) $\phi(x^{(k+1)}) < \phi(x^{(k)})$. Moreover, because the Rayleigh–Ritz theorem also gives$\varrho_{n+1}^{(k)} = \min_{x\in R^n}(\phi(x) + \lambda_L^{(k)} g(x))$, we have

$$\begin{aligned}\varrho_{n+1}^{(k)} &= \phi(x^{(k)}) + \lambda_L^{(k)} g(x^{(k)}) \\ &< \phi(x^{(k+1)}) + \lambda_L^{(k)} g(x^{(k+1)}) \\ &< \phi(x^{(k+1)}) + \lambda_L^{(k+1)} g(x^{(k+1)}) \\ &= \varrho_{n+1}^{(k+1)}.\end{aligned}$$

3. The proof of this case follows equivalently.
4. This is immediate from the definition of Problem 2.4. ∎

REMARK 2.2. *For an initial $0 < \lambda_L^{(0)} < \theta^c$, the tendency of the generated monotonic sequence for $\lambda_L^{(k)}$ depends on whether $\lambda_L^{(0)} < \lambda_L^*$ or $\lambda_L^{(0)} > \lambda_L^*$, but in either case $B(\lambda_L^{(k)})$ is always positive definite.*

THEOREM 2.5. *The iteration* (2.26) *with an initial $\lambda_L^{(0)} > 0$ converges to the unique solution, λ_L^*, of Problem* 2.4.

Proof. By Lemma 2.6 $\{\lambda_L^{(k)}\}$ is monotonic and by Lemma 2.5, statement 3, it is bounded by λ_L^*. Thus it converges. Suppose that it converges to the limit point $\tilde{\lambda}_L$; then this limit point should satisfy (2.26) in the limit, and $g(x_{\tilde{\lambda}_L}) = 0$. But now Problem 2.4 has a unique solution and thus $\tilde{\lambda}_L \equiv \lambda_L^*$. ∎

2.3. Algorithms. The theoretical results justify the basic algorithm for the solution x_{RTLS} of (1.2) which uses exact determination of the smallest eigenvalue for each update of the Lagrange parameter λ_L with RQI.

ALGORITHM 1 (EXACT RTLS: Alternating iteration on λ_L and x). *For given δ and initial guess $\lambda_L^{(0)} > 0$ calculate the eigenpair determined by $(\varrho_{n+1}^{(0)}, x^{(0)})$. Set $k = 0$. Update $\lambda_L^{(k)}$ and $x^{(k)}$ until convergence.*

1. While not converged
   Do
   (a) $\iota^{(k)} = 1$
   (b) **Inner Iteration:** Until sign condition is satisfied **Do:**
      i. Update $\lambda_L^{(k+1)}$ by (2.26).
      ii. Calculate the smallest eigenvalue, $\varrho_{n+1}^{(k)}$, and the corresponding eigenvector, $[x^{(k+1)}, -1]$, of matrix $\mathbf{B}(\lambda_L^{(k)})$.
      iii. If sign condition $g(x^{(k+1)})g(x^{(0)}) > 0$ is not satisfied, set $\iota^{(k)} = \iota^{(k)}/2$ else **Break**.
      End Do
   (c) Test for convergence. If converged **Break** else $k = k + 1$.
   End Do.
2. $x_{RTLS} = x^{(k)}$.

At the inner iteration in Algorithm 1 we find the minimum eigenvalue using an application of the approach presented in [2], based on cubically convergent RQI for the constant matrix $B(\lambda_L^{(k)})$. Block Gaussian elimination is used to improve the efficiency. Specifically, for fixed λ_L we iterate over j such that at iteration j we wish to find the

vector $y^{(k,j+1)} = ((x^{(k,j+1)})^T, -1)^T$ such that

$$\mathbf{B}(\lambda_L^{(k)})y^{(k,j+1)} = \beta_{(k,j)}y^{(k,j)}, \tag{2.27}$$

$$\mathbf{B}(\lambda_L^{(k)}) = \begin{pmatrix} J^{(k,j)} & A^Tb \\ b^TA & \eta_{(k,j)} \end{pmatrix}, \tag{2.28}$$

$$J^{(k,j)} = A^TA + \lambda_L^{(k)}L^TL - \rho_{(k,j)}I_n, \qquad \eta_{(k,j)} = b^Tb - \lambda_L^{(k)}\delta^2 - \rho_{(k,j)}, \tag{2.29}$$

where $\rho_{(k,j)}$ is the RQI shift. Here we use the double index (k, j) to indicate that the inner iteration to find the eigenvalue is over index j as compared to the outer iteration for λ_L which is over k. Having made this distinction, we now assume the dependence on k whenever iteration j is denoted. We suppose that the matrix $J^{(j)}$ is positive definite, certainly the case without shift by assumption on the initial choice of $\lambda_L^{(0)}$, so that we can apply block Gaussian elimination

$$\begin{pmatrix} J^{(j)} & A^Tb \\ 0 & \tau_j \end{pmatrix}\begin{pmatrix} x^{(j+1)} \\ -1 \end{pmatrix} = \beta_j \begin{pmatrix} x^{(j)} \\ -(z^{(j)})^Tx^{(j)} - 1 \end{pmatrix},$$

where

$$\tau_j = \eta_j - b^TAz^{(j)} \tag{2.30}$$

and $x^{(j+1)} = z^{(j)} + \beta_j u^{(j)}$. Here $z^{(j)}$ and $u^{(j)}$ solve the systems

$$J^{(j)}z^{(j)} = A^Tb, \tag{2.31}$$

$$J^{(j)}u^{(j)} = x^{(j)}, \tag{2.32}$$

and the scaling parameter is given by

$$\beta_j = \tau_j/((z^{(j)})^Tx^{(j)} + 1). \tag{2.33}$$

REMARK 2.3. *The algorithm as presented to match the theoretical results requires precise determination of the smallest eigenvalue for each λ_L. However, an inexact determination, particularly in early stages of the iteration, may increase efficiency by reducing the total number of iterations. Moreover, the key requirement of the convergence result is that the update $x_{\theta^{(k+1)}}$ is such that the sign property for function g is maintained. Thus, suppose that we do not solve the eigenproblem exactly for each $\lambda_L^{(k)}$, but that instead an approximate eigenpair is found, $(\tilde{\rho}_{n+1}^{(k)}, \tilde{x}_{\theta^{(k+1)}})$, for which $g(\tilde{x}_{\theta^{(k+1)}})$ maintains the sign condition; then the iteration will still converge. This leads to modification of Algorithm* 1 *based on inexact update for the eigenvalue.*

ALGORITHM 2 (INEXACT RTLS: Alternating iteration on λ_L and x). *Implement the exact algorithm but initialized with $0 < \lambda_L^{(0)} < \theta^c$ chosen such that the initial matrix $B(\lambda_L^{(0)})$ is positive definite. At each iteration do not search for the exact eigenpair for each k, rather use inverse iteration and seek satisfactory $x^{(k,J_k)}$, such that $g(x^{(k,J_k)})g(x^{(0)}) > 0$. If this condition is satisfied for $j = J_k$, assign $x^{(k)} = x^{(k,J_k)}$ and update $\lambda_L^{(k+1)}$. The initial vector for each inner inverse iteration is $x^{(k,0)} = x^{(k-1,J_{k-1})}$.*

REMARK 2.4. *It is immediate to see from the convergence theory that if the requirement on the sign of g is relaxed, a divergent sequence can result. It was this version of Algorithm* 2 *that was implemented in* [6]. *In particular, without the condition on the sign of g, each inner iteration to calculate $x^{(k)}$ uses just one step, $j = 1$, and the matrix $\boldsymbol{B}(\lambda_L^{(k)})$ is updated each step.*

REMARK 2.5. *In Algorithms* 1 *and* 2 *we assume that* $\gamma = \delta^2$, *where* $\lambda_L^{(0)}$ *is required. While the theory does not immediately follow, these algorithms can be modified to use* $\gamma = \|Lx^{(k,j)}\|^2$, *where initial solution* $x^{(0)}$ *or* $x^{(0,0)}$ *is required. This modification introduces new versions of both algorithms, which we denote by* 1.2 *and* 2.2, *resp., reserving the notation* 1.1 *and* 2.1 *for the former versions. If blow-up does not occur, which we demonstrate through our numerical experiments is seldom the case, we find that the iteration converges much more quickly.*

REMARK 2.6. *While the exact determination of the smallest eigenvalue at any step will be made efficient if the shift of the RQI is utilized, it should be clear that it is not, in general, desirable to use the shift when we pose the problem in the inexact form, for which we want to change* λ_L *efficiently to get to the RTLS solution, rather than to find each intermediate eigenvalue precisely. Thus, in general, given an initial choice of* λ_L *such that* $\boldsymbol{B}$ *is positive definite, the block matrix* $J^{(j)}$ *without shift is guaranteed positive definite.*

2.4. Interdependence of parameters. In the preceding algorithms we assume that the physical parameter δ is known a priori, which may not always be the case. Hence we need to understand the relationship between δ and the other parameters λ_L and λ_I in order to determine an algorithm for which δ is not provided.

Consistent with earlier notation, we distinguish the solution of the RTLS problem via the δ-specified algorithm as x_δ. Moreover, we use $J(\lambda_L) = A^TA - \phi(x_{\lambda_L})I + \lambda_L L^TL$ and $s(x_{\lambda_L}) = A^TAx_{\lambda_L} - A^Tb - \phi(x_{\lambda_L})x_{\lambda_L}$ for which $J(\lambda_L)x_{\lambda_L} = A^Tb$ and $s(x_{\lambda_L}) = -\lambda_L L^TLx_{\lambda_L}$.

THEOREM 2.6. *Suppose matrix* $J(\lambda_L)$ *is positive definite and* $\lambda_L > 0$; *then*

1. $\frac{d\phi(x_{\lambda_L})}{d\lambda_L} > 0$, $\phi(x_{\lambda_L})$ *is monotonically increasing with respect to* λ_L, *and*
2. $\frac{d(\|Lx_{\lambda_L}\|^2)}{d\lambda_L} < 0$, $\|Lx_{\lambda_L}\|^2$ *is monotonically decreasing with respect to* λ_L.

Proof. Differentiating $J(\lambda_L)x_{\lambda_L} = A^Tb$ with respect to λ_L yields

$$J(\lambda_L)\frac{dx_{\lambda_L}}{d\lambda_L} = \left(\frac{d\phi(x_{\lambda_L})}{d\lambda_L}I - L^TL\right)x_{\lambda_L}.$$

Now

$$\begin{aligned}\frac{d\phi(x_{\lambda_L})}{d\lambda_L} &= (\nabla_{x_{\lambda_L}}\phi(x_{\lambda_L}))^T\frac{dx_{\lambda_L}}{d\lambda_L}\\ &= \frac{2}{1+\|x_{\lambda_L}\|^2}s^T(x_{\lambda_L})\frac{dx_{\lambda_L}}{d\lambda_L}\\ &= -\frac{2\lambda_L x_{\lambda_L}^TL^TL}{1+\|x_{\lambda_L}\|^2}\frac{dx_{\lambda_L}}{d\lambda_L}. \qquad (2.34)\end{aligned}$$

Rearranging yields

$$\frac{d\phi(x_{\lambda_L})}{d\lambda_L}\left(\frac{1+\|x_{\lambda_L}\|^2}{2}\right) = \lambda_L x_{\lambda_L}^TL^TL\left(\frac{d\phi(x_{\lambda_L})}{d\lambda_L}J(\lambda_L)^{-1}x_{\lambda_L} - J(\lambda_L)^{-1}L^TLx_{\lambda_L}\right).$$

Hence

$$\frac{d\phi(x_{\lambda_L})}{d\lambda_L}\left(\frac{1+\|x_{\lambda_L}\|^2}{2} + \lambda_L x_{\lambda_L}^TL^TLJ(\lambda_L)^{-1}x_{\lambda_L}\right) = \lambda_L x_{\lambda_L}^TL^TLJ(\lambda_L)^{-1}L^TLx_{\lambda_L} > 0$$

by assumptions on $J(\lambda_L)$ and λ_L, and the first statement follows immediately.

On the other hand,

$$\frac{d(\|Lx_{\lambda_L}\|^2)}{d\lambda_L} = 2x_{\lambda_L}^T L^T L \frac{dx_{\lambda_L}}{d\lambda_L},$$

which, after substitution in (2.34), gives

$$\frac{d\phi(x_{\lambda_L})}{d\lambda_L} = -\frac{\lambda_L}{(1+\|x_{\lambda_L}\|^2)} \frac{d(\|Lx_{\lambda_L}\|^2)}{d\lambda_L},$$

and the second statement follows. □

These results justify the introduction of alternative algorithms.

2.4.1. Bisection search. Because of the direct monotonic relationship between parameters $\delta = \|Lx_{\lambda_L}\|$ and λ_L, we can use a standard bisection search technique on parameter λ_L to obtain an update mechanism for λ_L. With this approach the number of solves for each λ_L is determined by the precision required and the initial interval for bisection and is thus of most use in situations for which we know that the class of problems is difficult to solve. This gives the following algorithm, for which details are standard.

ALGORITHM 3 (RTLS: Bisection search on λ_L). *Given δ, a search tolerance* **TOL** *on the active constraint, $|\|Lx_{\lambda_L}\| - \delta| \le$ **TOL**, and two initial choices of λ_L for which $g(x_{\lambda_L})$ are of different signs, do bisection until the tolerance is satisfied. At each iteration estimate solution $\hat{x}_{\lambda_L}$ by Algorithm 2 with γ updated each step, namely Algorithm 2.2 except λ_L is fixed.*

2.4.2. An L-curve algorithm. The earlier algorithms assume a priori information to designate δ which may not be available. We consider instead, then, an approach based on the use of the L-curve [8, 10] to give a δ-independent algorithm for the formulation

$$\min_x \phi(x) + \mu\|Lx\|^2. \tag{2.35}$$

Here the positive regularization parameter μ controls how much weight is given to the penalty function $\|Lx\|^2$ as compared to the Rayleigh quotient $\phi(x)$. Necessary conditions for a minimum of (2.35) are the same as for (2.2) except for (2.9). If the constraint is active, the solution satisfies

$$(A^TA + \lambda_I I + \mu(1+\|x\|^2)L^TL)x = A^Tb,$$

where $\lambda_I = -\phi(x)$. Substituting $\lambda_L = \mu(1+\|x\|^2)$, we once again obtain (2.5):

$$(A^TA + \lambda_I I + \lambda_L L^TL)x = A^Tb.$$

For each fixed parameter λ_L, the solution x_{λ_L} is equivalent to the x_{RTLS} solution obtained with constraint parameter $\delta = \|Lx_{\lambda_L}\|$. Hence we need to determine λ_L so that it simultaneously gives a small Rayleigh quotient $\phi(x_{\lambda_L})$ and a moderate value of the penalty term $\|Lx_{\lambda_L}\|^2$. We use the L-curve method which was designed for the Tikhonov regularized LS problem [8, 10] for the *log-log* scale plot of $\phi(x_{\lambda_L})$ versus $\|Lx_{\lambda_L}\|^2$.

ALGORITHM 4 (RTLS: L-curve). *Given a discrete set of values for λ_L on an interval $[a, b]$, find RTLS solutions x_{λ_L}. Generate the L-curve and pick the lower left corner point of the curve to generate x_{RTLS}.*

1. **For** λ_L over a discrete set.
   Inner iteration For fixed λ_L, calculate RTLS solution x_{λ_L} by alternatively updating x_{λ_L} by solving (2.5) and λ_I through (2.7) until the inner iteration has converged.
   End For
2. Plot on *log-log* scale the pairs $\phi(x_{\lambda_L})$ versus $\|Lx_{\lambda_L}\|^2$.
3. Find the lower left corner point of the L-curve, the corresponding parameter λ_L, and solution $x_{\lambda_L} = x_{RTLS}$.

REMARK 2.7. *To carry out the final step of the algorithm we could use "Algorithm FindCorner" in* [10].

3. Computational considerations.

3.1. Termination criteria. In the inner iterations for the Rayleigh quotient or inverse iteration, where also the system matrix may depend on the current update if γ is updated each step, we test convergence on the residual $r^{(j)} = B(x^{(j)})\bar{y}^{(j)} + \lambda_I(x^{(j)})\bar{y}^{(j)}$, where $\bar{y}^{(j)}$ is $y^{(j)}$ normalized. It is easy to verify that

$$(B(x^{(j)}) + C^{(j)})\bar{y}^{(j)} = -\lambda_I(x^{(j)})\bar{y}^{(j)},$$

where $C^{(j)} = -r^{(j)}(\bar{y}^{(j)})^T$. Let ϵ represent machine accuracy and c be a quite mild function of degree $n+1$; then the best accuracy we can expect to achieve is $\|C^{(j)}\|/\|B(x^{(j)})\| \leq c\epsilon$ [17, Chap. 5, sect. 58]. Hence

$$\|C^{(j)}\| = \|r^{(j)}\| \leq c\epsilon\|B(x^{(j)})\| \approx c\epsilon\|[A,b]\|^2.$$

Since $B(x^{(j)})$ is a symmetric matrix, the accuracy of $\lambda_I(x^{(j)})$ is also approximately $c\epsilon\|[A,b]\|^2$. This suggests using $|\lambda_I^{(j)} - \lambda_I^{(j-1)}|/|\lambda_I^{(j)}| < \mathbf{TOL}$ as stopping criterium, where **TOL** is a tolerance. $\frac{\|r^{(j)}\|}{|\lambda_I^{(j)}|} < \mathbf{TOL}$ may also be used as termination criterium.

When δ is known we may directly use $|\|Lx^{(j)}\| - \delta|$ as stopping criterium. Also $\|r^{(j)}\|$ is a measurement of the distance of $x^{(j)}$ to the boundary (2.4). In fact, by the Cauchy–Schwarz inequality, $\|\bar{y}^{(j)}\| = 1$, and using, from (2.9), μ as a function of x,

$$\begin{aligned} \|r^{(j)}\| &\geq |(\bar{y}^{(j)})^T(B(x^{(j)}) + \lambda_I(x^{(j)})I)\bar{y}^{(j)}| \\ &= |\mu(x^{(j)})(\|Lx^{(j)}\|^2 - \delta^2)|. \end{aligned} \tag{3.1}$$

Thus, the residual $\|r^{(j)}\|$ also provides an upper estimate for the violation of the constraint condition (2.4) and, if $\bar{y}^{(j)}$ is sufficiently close to an eigenvector of $B(x^{(j)})$, then the inequality in (3.1) is close to an equality. Since we solve the eigenproblem for B and need to find $-\lambda_I$, we would expect $(B+\lambda_I^{(j)}I)\bar{y}^{(j)}$ becomes zero if $(-\lambda_I^{(j)}, \bar{y}^{(j)})^T$ is an eigenpair for B.

3.2. The generalized SVD (GSVD) of $[A, L]$. All of the presented algorithms depend on the efficiency of solving systems with coefficient matrix $A^TA + \lambda_L L^TL$, or the shifted version $A^TA + \lambda_L L^TL - \rho_k I$. Here we focus on the derivation of an efficient algorithm for the solution of systems with system matrix, J, without shift. Notice that, without loss of generality, we drop the dependence on iteration (k,j), and consider the solution of the system

$$(A^TA + \lambda_L L^TL)w = f. \tag{3.2}$$

While different approaches can be considered for (3.2), we also note the similarity of (3.2) with the system to be solved in Tikhonov regularization of the least squares

TABLE 3.1
Algorithmic summary and comparison.

	Algorithm $*.1$	Algorithm $*.2$	Bisection	L-curve
δ	given	given	given	unknown
$x^{(0)}$	random	required	required	required
$\lambda_L^{(0)}$	given	derived from $x^{(0)}$	derived from $x^{(0)}$	derived from $x^{(0)}$
subalg.	No	No	Algorithm 2.2	Algorithm 2.2

problem. Hence we should use the algorithms which have been demonstrated as successful for regularized LS. Moreover, we can safely assume that matrix L, which in our examples is a low order derivative operator, is well conditioned. In particular, the smallest nonzero singular values of the first and second order derivative operators are of order n^{-1} and n^{-2}, resp., and their null spaces are spanned by very *smooth* vectors. Thus, if λ_L is not too small, matrix $A^TA + \lambda_L L^TL$ is well posed for a large class of matrices A, and the GSVD of the matrix pair $[A, L]$ can be calculated with a stable numerical method [7]. This approach, used to solve (3.2) [7], also motivates use of algorithms without shift. Using the algorithm of Bai and Demmel [1], the calculation of the GSVD for matrix pair $[A, L]$ requires $2m^2n + 15n^3$ flops (the coefficient of n^3 depends on the number of iterations required). Given the GSVD the solution of each equation (3.2) costs just $8n^2$ flops.

3.3. Summary of the algorithms. In the preceding sections we have presented several different algorithmic approaches for the solution of the given RTLS problem. We now summarize these algorithms with respect to the initialization requirements and the subalgorithm that is used to solve an eigenproblem with fixed parameter λ_L. We list the requirements in Table 3.1, where Algorithm $*.1$ and Algorithm $*.2$ represent versions $\gamma = \delta^2$ and $\gamma = \|Lx^{(k)}\|^2$, resp.

4. Numerical experiments. To test the given algorithms we mainly use test examples taken from Hansen's *Regularization Tools* [9]. Three functions, *ilaplace*, *phillips*, and *shaw*, are used to generate matrices A, right-hand sides b, and solutions $x^\sharp$ so that $Ax^\sharp = b$ is satisfied. In all cases, the data are scaled so that $\|A\|_F = \|Ax^\sharp\|_2 = 1$, and a 5% Gaussian perturbation is added to both coefficient matrix and right-hand side. For *ilaplace* and *shaw* matrix A has size 65×64, and for *phillips* the matrix is 64×64 [3, 9]. We let operator $L \in R^{(n-1)\times n}$ approximate the first-derivative operator. For algorithms in which δ is specified, we choose $\delta = 0.9\|Lx^\sharp\|$. In all tests we choose tolerance $\mathbf{TOL} = 10^{-4}$, and we denote the estimated solution of each algorithm by x_{est}.

In the results we report the relative error with respect to $x^\sharp$. On the other hand, we know the solutions should converge to x_{RTLS}, which is the solution of the equation subject to constraint. Thus we may expect that evaluation compared to $x^\sharp$ is limited for a single test, and that it is the speed with which a converged solution satisfying the constraint is achieved, which is important. Thus in Test 4.3 we repeat tests over 100 perturbations for each experiment and report the average results for each case, except for the L-curve in which we present results of one sample perturbation. We measure the speed with respect to the numbers of system solves of type (3.2) that are required, hence providing a comparison between algorithms. To give the total cost of each test, we add the cost for the GSVD and the iterations, i.e., $2m^2n + 15n^3 + K \cdot 8n^2$ flops, where K is the number of solves, and report the number of megaflops. In each case we initialize the iteration with $\lambda_L^{(0)} = 0.1$, and for Algorithm 2.2 with $x^{(0,0)} = x_{RLS}$ obtained with regularization parameter $\lambda = .001$.

For the test of the L-curve algorithm we pick 20 equally spaced points, with respect to the log scale, on the interval $[a, b] = [1.0e-6, 0.1]$. For any choice of λ_L we stop the inner iteration if convergence is not achieved in 15 steps. If the curve has a clear L-shape, 20 points are sufficient to identify the corner because more points are located near the corner than at other places on the curve.

Test 4.1 (evaluation of inexact and exact algorithms). Here we demonstrate the impact of use of the exact solve by Algorithm 1 as compared to the inexact approach in Algorithm 2, in which for the inexact solve we search only for a new update which satisfies the sign condition. We find that primarily $J_k = 1$; namely the inexact solve mostly uses one step of inverse iteration prior to update of parameter λ_L. The long-term and short-term convergence history for $-\lambda_I^{(k)}$ is illustrated in Figure 4.1. In these tests we do not update γ occurring in $\mathbf{B}$, but fix $\gamma = \delta^2$. This test is thus a true comparison for the convergence theory for inexact solve in place of the exact solve. Clearly, the costs for the inexact solve are less than the total cost for determination of the exact eigenpair at each outer iteration, but the total impact depends on the algorithm used for the exact solve. Thus we do not report relative costs in each case. We observe that over the long term there is no detrimental impact on the convergence behavior, even though we see that at the early iterations the solutions obtained are not exactly the same. It is clear that inexact solve produces an alternative update in the early steps without being deleterious for ultimate convergence.

Test 4.2 (evaluation of inclusion of RQ shift). We now consider the impact of the use of the shift for improving the convergence of the inexact algorithm, Algorithm 2. In Figure 4.2 we show the lack of impact on the convergence of inclusion of the shift for the inner iteration of Algorithm 2. The top and bottom three figures are associated with Algorithms 2.1 and 2.2, resp. We illustrate three cases, the first without any shift, the second in which we shift at all steps, and the third in which, consistent with the RQI for the TLS problem introduced by [2], we shift after the first step. We

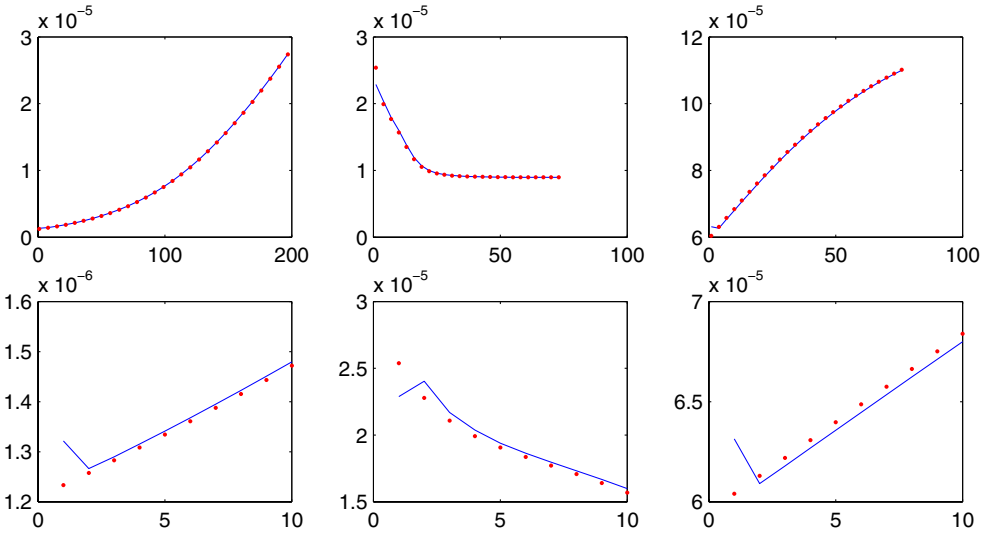


FIG. 4.1. *The figures compare the convergence history for* $-\lambda_I^{(k)}$ *for the exact algorithms compared to the inexact. The dotted and dashed lines show the convergence for the exact and inexact algorithms, resp. The first row shows the whole convergence history while the second row shows the first* 10 *steps. From left to right, examples* ilaplace, shaw, *and* phillips, *resp.*

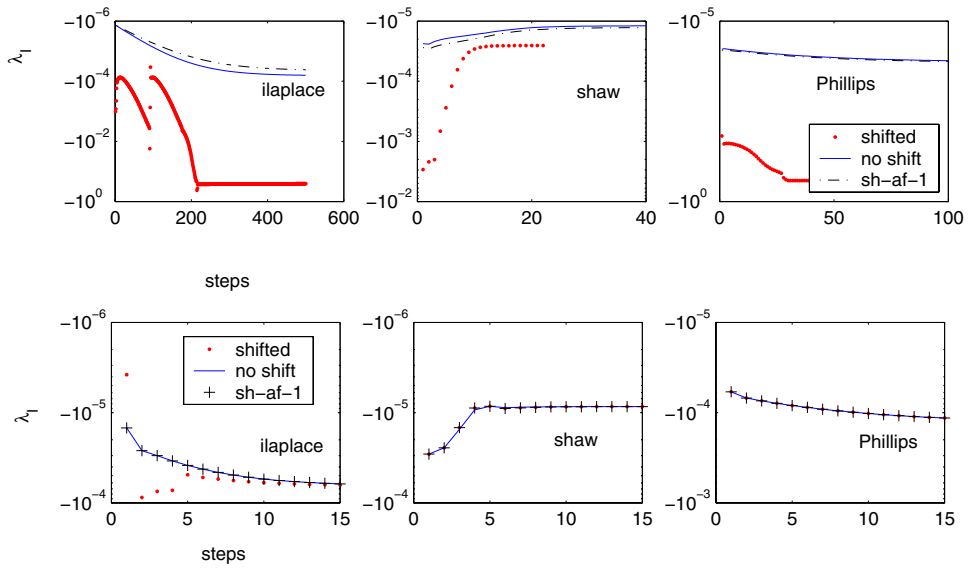


FIG. 4.2. *The figures show the convergence history of* $-\lambda_I$ *for Algorithms* 2.1 *and* 2.2*, top and bottom, resp., with the shift added at different stages in the iteration process, i.e., shifted for all steps ("shifted"), no shift at all ("no shift"), and the shift added after the first step ("sh-af-*1*"). From left to right, examples* ilaplace*,* shaw*, and* phillips*, resp.*

note the different line type used for the case when the shift is added after the first step between the upper and lower figures. This is deliberate because, in addition, Algorithm 2.2 converges with far fewer iterations (compare the scales on the x-axes) and so the use of + also for Algorithm 2.1, which requires many iterations, would mask all the other results in these figures.

It is clear that adding the shift at every step can cause Algorithm 2.1 to converge to an eigenpair which is not the smallest as defined by the eigenvalue. This possibility exists because the algorithm is initialized with a random vector which then generates a bad initial RQ shift. This problem is avoided if the shift occurs only after one step of inverse iteration, and the results are almost the same as without shift for all steps—both approaches converge to the RTLS solution. On the other hand, Algorithm 2.2 always converges to the RTLS solution, and shift does not cause any significant difference in the convergence history of λ_I after the first few steps of the iteration.

In summary, adding the shift makes no positive contribution to the convergence, contrary to the case for RQI for the TLS problem. Moreover, with inclusion of the shift we cannot take advantage of the calculation of the GSVD for the augmented matrix $[A, L]$. Thus our results demonstrate no reason to include the shift in (2.29), which also further justifies our assumption that matrix J remains positive definite throughout the iteration.

Test 4.3 (comparison of the algorithms). Here we emphasize the improvement due to setting $\gamma = \|Lx^{(k,j)}\|^2$ in the right corner of $B(x^{(k,j)})$ as compared with $\gamma = \delta^2$. Details of average results for all four algorithms, over 100 perturbations, are provided in Tables 4.1–4.3, and the solutions of one sample perturbation are illustrated in Figure 4.3. In the tables, the relative error reported is the average relative error of x_{est} to $x^\sharp$.

TABLE 4.1
Average solutions for ilaplace (*Test* 4.3).

Algorithm	λ_L	λ_I	K	Cost mflops	Relerr
1.1 $\gamma = \delta^2$	6.6382e-01	−5.6144e-05	799	NA	6.75e-02
1.2 $\gamma = \|Lx^{(k,j)}\|^2$	6.6433e-01	−5.6228e-05	54.4	NA	6.75e-02
2.1 $\gamma = \delta^2$	6.6382e-01	−5.6144e-05	799	31.0	6.75e-02
2.2 $\gamma = \|Lx^{(k,j)}\|^2$	6.6433e-01	−5.6228e-05	54.2	6.2	6.75e-02
Bisection	6.6231e-01	−5.5937e-05	100.6	7.8	6.79e-02
L-curve (sample)	5.6234e-04	−3.0242e-08	161	9.8	1.7470e-02

TABLE 4.2
Average solutions for shaw (*Test* 4.3).

Algorithm	λ_L	λ_I	K	Cost mflops	Relerr
1.1 $\gamma = \delta^2$	3.1329e-04	−9.9912e-06	98.1	NA	9.13e-02
1.2 $\gamma = \|Lx^{(k,j)}\|^2$	3.1296e-04	−9.9907e-06	21.0	NA	9.12e-02
2.1 $\gamma = \delta^2$	3.1310e-04	−9.9895e-06	99.4	7.7	9.11e-02
2.2 $\gamma = \|Lx^{(k,j)}\|^2$	2.9090e-04	−9.8089e-06	25.8	5.3	9.51e-02
Bisection	2.7939e-04	−1.0033e-05	81.5	7.1	9.46e-02
L-curve (sample)	1.7783e-04	−1.0094e-05	158	9.7	1.0478e-01

TABLE 4.3
Average solutions for phillips (*Test* 4.3).

Algorithm	λ_L	λ_I	K	Cost mflops	Relerr
1.1 $\gamma = \delta^2$	1.8454e-01	−1.3426e-04	368.6	NA	9.05e-02
1.2 $\gamma = \|Lx^{(k,j)}\|^2$	1.8454e-01	−1.3426e-04	71.5	NA	9.05e-02
2.1 $\gamma = \delta^2$	1.8454e-01	−1.3426e-04	369.0	17.0	9.05e-02
2.2 $\gamma = \|Lx^{(k,j)}\|^2$	1.8454e-01	−1.3426e-04	71.6	6.8	9.05e-02
Bisection	1.8502e-01	−1.3460e-04	66.3	6.6	9.05e-02
L-curve (sample)	5.6234e-04	−4.4807e-06	119	8.36	5.1365e-02

We note that the total numbers of outer iterations for Algorithm 1 and Algorithm 2 are comparable, thus again demonstrating the benefit of the use of the inexact solve for each outer iteration. Again we do not report the costs of the exact solve, denoted in the tables by NA, which depends on the chosen algorithm and is certainly not optimal if inverse iteration is used. Given the lack of benefit of the use of exact solve, we chose not to investigate the most efficient technique for its solution. In all cases, we see a dramatic decrease in the total number of steps required to reach convergence for Algorithm 2.2 as compared to Algorithm 2.1. While the solutions are different in all cases, because of the dependence on the specific converged value for λ_L, all solutions other than those obtained by the L-curve algorithm are qualitatively similar; see the figures on the left of Figure 4.3. We note that example *shaw* does not give a good L-shape and thus it is hard to determine the optimal λ_L.

Test 4.4 (comparison with solution based on the quadratic eigenvalue problem (QEP) [14]). To compare the approach with that using the QEP we compare Algorithm 2.1 with the QEP again over 100 cases, each with the random 5% perturbation. For both algorithms we adopt the stopping rule $\|x^{(k+1)} - x^{(k)}\|/\|x^{(k+1)}\| < \mathbf{TOL}$ used in [14], and the QEP program is written exactly as stated for **rtlsqep** in [14]. We use a random initial solution $x^{(0)}$ and matrix-vector multiplication to avoid matrix multiplication. Algorithm 2.1 is initialized in each case with $\lambda_L = 0.1$. In some situations—for example, *ilaplace*—this generates an apparently *zero* cost solution. Actually this corresponds to a one step iteration to convergence because the initial-

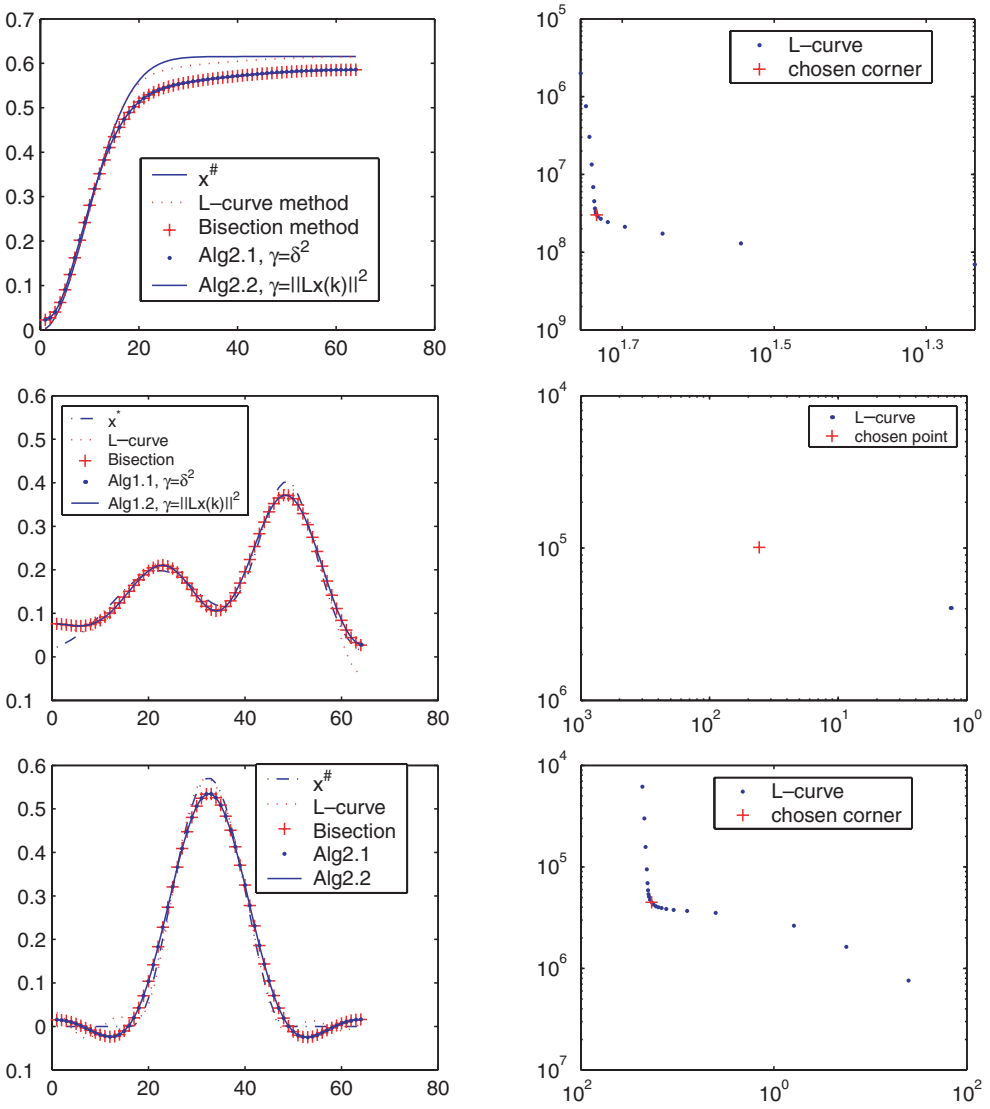


FIG. 4.3. *From top to bottom, examples* ilaplace, phillips, *and* shaw, *resp. Solutions are indicated on the left and the L-curve on the right.*

ization $\lambda_L = 0.1$ presents an almost perfect estimation to the regularization parameter λ_L, which would *never* occur for use of random $x^{(0)}$ with QEP.

The results are summarized in Figure 4.4, which shows the distribution of relative errors of x_{est} to $x^\sharp$ (two top rows of figures), $-\lambda_I(x_{est})$ (middle two rows of figures), and the CPU costs in seconds (last two rows of figures). In each case the figures are organized with results for Algorithm 2.1 first, followed by those for QEP, and with, from left to right, examples *ilaplace*, *shaw*, and *phillips*, resp.

For *ilaplace* Algorithm 2.1 has generally smaller error but is a little more expensive than QEP, while the results with *shaw* are similar but Algorithm 2.1 is cheaper, and *phillips* outperforms the QEP in all measures.

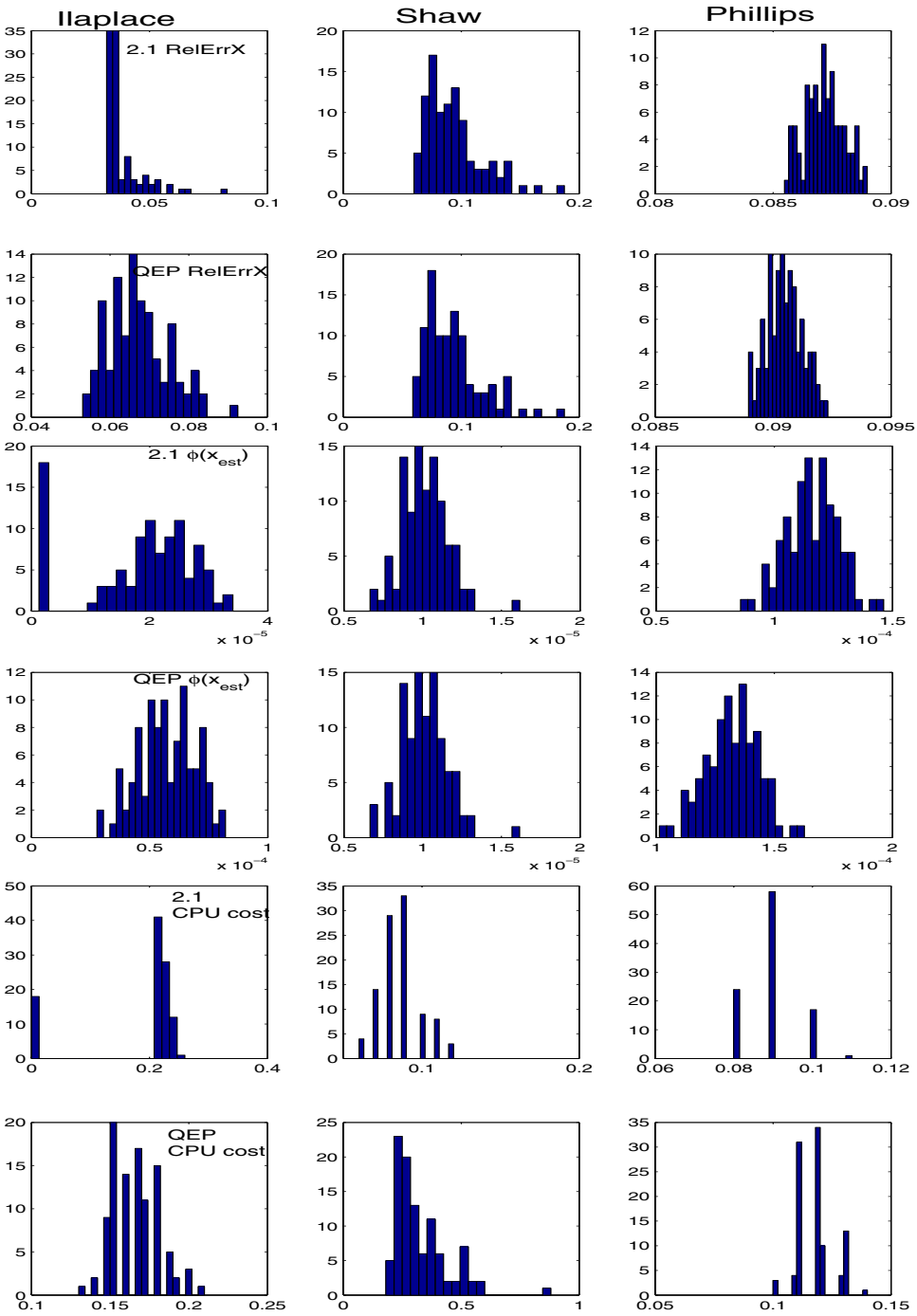


FIG. 4.4. *Comparison of Algorithm* 2.1 *and the QEP algorithm, Test* 4.4.

5. Conclusions. We have demonstrated new algorithms for the solution of the RTLS problem. These algorithms employ the relationship between the RTLS solution

and the eigensolution of an augmented matrix. The results are summarized as follows:

1. In the case that a good estimate on the constraint condition for the solution is available, an efficient approach uses inverse iteration for the solution of the eigenproblem combined with the GSVD for solution of the systems arising at each inverse iteration.
2. If a good estimate for the constraint parameter is available, but the algorithm is shown to converge slowly for a given class of problems, bisection search may be used to predict the total number of outer steps required for a given desired accuracy.
3. For cases without constraint information, the L-curve approach used for regularized LS has been adapted for regularized TLS.

Numerical experiments have been presented which verify all algorithms and we conclude with the following.

1. Algorithm 2.2 provides an efficient and practical approach for the solution of the RTLS problem in which a good estimate of the physical parameter is provided.
2. Otherwise, if blow-up occurs, bisection search may yield a better solution satisfying the constraint condition.
3. If no constraint information is provided, the L-curve technique can be successfully employed.
4. Algorithm 2.1 performs better than QEP for all of our tests.

In all cases we have demonstrated a constructive and practical approach for the solution of RTLS problems.

Acknowledgment. The authors gratefully acknowledge the comments of three anonymous referees who suggested that we seek a proof of the convergence of our basic algorithm, which ultimately led to our improvement of the reliability of the solution technique.

REFERENCES

[1] Z. Bai and J. Demmel, *Computing the generalized singular value decomposition*, SIAM J. Sci. Comput., 14 (1993), pp. 1464–1486.

[2] A. Björck, P. Heggernes, and P. Matstoms, *Methods for large scale total least squares problems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 413–429.

[3] 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.

[4] G. H. Golub and C. Van Loan, *An analysis of the total least squares problem*, SIAM J. Numer. Anal., 17 (1980), pp. 883–893.

[5] G. H. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[6] H. Guo and R. A. Renaut, *A regularized total least squares algorithm*, in Total Least Squares and Errors-in-Variables Modeling: Analysis, Algorithms and Applications, S. Van Huffel and P. Lemmerling, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002, pp. 57–66.

[7] P. C. Hansen, *Regularization, GSVD and truncated GSVD*, BIT, 29 (1989), pp. 491–504.

[8] P. C. Hansen, *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Rev., 34 (1992), pp. 561–580.

[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 and D. P. O'Leary, *The use of the L-curve in the regularization of discrete ill-posed problems*, SIAM J. Sci. Comput., 14 (1993), pp. 1487–1503.

[11] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[12] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects*

and Analysis, SIAM, Philadelphia, 1991.
[13] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[14] D. Sima, S. Van Huffel, and G. H. Golub, *Regularized Total Least Squares Based on Quadratic Eigenvalue Problem Solvers*, Tech. Report SCCM-03-03, SCCM, Stanford University, Stanford, CA, 2003.
[15] A. N. Tikhonov, *Solution of incorrectly formulated problems and the regularization method*, Soviet Math. Dokl., 4 (1963), pp. 1035–1038.
[16] A. N. Tikhonov and V. Y. Arsenin, *Solution of Ill-Posed Problems*, John Wiley & Sons, New York, 1977.
[17] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 2, pp. 477–497

A JACOBI–DAVIDSON TYPE METHOD FOR THE TWO-PARAMETER EIGENVALUE PROBLEM*

MICHIEL E. HOCHSTENBACH†, TOMAŽ KOŠIR‡, AND BOR PLESTENJAK‡

Abstract. We present a new numerical method for computing selected eigenvalues and eigenvectors of the two-parameter eigenvalue problem. The method does not require good initial approximations and is able to tackle large problems that are too expensive for methods that compute all eigenvalues. The new method uses a two-sided approach and is a generalization of the Jacobi–Davidson type method for right definite two-parameter eigenvalue problems [M. E. Hochstenbach and B. Plestenjak, *SIAM J. Matrix Anal. Appl.*, 24 (2002), pp. 392–410]. Here we consider the much wider class of nonsingular problems. In each step we first compute Petrov triples of a small projected two-parameter eigenvalue problem and then expand the left and right search spaces using approximate solutions to appropriate correction equations. Using a selection technique, it is possible to compute more than one eigenpair. Some numerical examples are presented.

Key words. two-parameter eigenvalue problem, subspace method, Jacobi–Davidson method, correction equation, Petrov–Galerkin, two-sided approach

AMS subject classifications. 65F15, 15A18, 15A69

DOI. 10.1137/S0895479802418318

1. Introduction.

We are interested in computing one or more eigenpairs of the *two-parameter eigenvalue problem*

$$\begin{aligned} A_1x_1 &= \lambda B_1x_1 + \mu C_1x_1, \\ A_2x_2 &= \lambda B_2x_2 + \mu C_2x_2, \end{aligned} \tag{1.1}$$

where A_i, B_i, and C_i are given $n_i \times n_i$ matrices over $\mathbb{C}$, $\lambda, \mu \in \mathbb{C}$ and $x_i \in \mathbb{C}^{n_i}$ for $i = 1, 2$. A pair (λ, μ) is called an *eigenvalue* if it satisfies (1.1) for nonzero vectors x_1, x_2. The tensor product $x_1 \otimes x_2$ is then the corresponding *right eigenvector*. Similarly, $y_1 \otimes y_2$ is the corresponding *left eigenvector* if $0 \neq y_i \in \mathbb{C}^{n_i}$ and $y_i^*(A_i - \lambda B_i - \mu C_i) = 0$ for $i = 1, 2$.

Multiparameter eigenvalue problems of this kind arise in a variety of applications [1], particularly in mathematical physics when the method of separation of variables is used to solve boundary value problems [23]. When the separation constants cannot be decoupled, two-parameter Sturm–Liouville problems of the form

$$-(p_i(x_i)y_i'(x_i))' + q_i(x_i)y_i(x_i) = (\lambda a_{i1}(x_i) + \mu a_{i2}(x_i))y_i(x_i), \tag{1.2}$$

where $x_i \in [a_i, b_i]$, with boundary conditions

$$\begin{aligned} y_i(a_i)\cos\alpha_i - y_i'(a_i)\sin\alpha_i = 0, \quad 0 \leq \alpha_i \leq \pi, \\ y_i(b_i)\cos\beta_i - y_i'(b_i)\sin\beta_i = 0, \quad 0 \leq \beta_i \leq \pi, \end{aligned}$$

*Received by the editors November 21, 2002; accepted for publication (in revised form) by Z. Strakoš April 8, 2004; published electronically January 12, 2005.
http://www.siam.org/journals/simax/26-2/41831.html

†Department of Mathematics, Case Western Reserve University, 10900 Euclid Avenue, Cleveland, OH 44106-7058 (hochsten@case.edu). The research of this author was supported in part by NSF grant DMS-0405387. Part of this research was done while the author was employed by Utrecht University.

‡Department of Mathematics, University of Ljubljana, Jadranska 19, SI-1000 Ljubljana, Slovenia (tomaz.kosir@fmf.uni-lj.si, bor.plestenjak@fmf.uni-lj.si). The research of these authors was supported in part by the Ministry of Education, Science, and Sport of Slovenia (Research Projects Z1-3136 and PO-0508).

can arise, where $\alpha_i \in [0, \pi)$, $\beta_i \in (0, \pi]$, and p_i', q_i, a_{i1}, a_{i2} are real valued and continuous functions, for $i = 1, 2$. Using discretization, the problem (1.2) can be converted into the form (1.1). As an example, let us consider the equation $\Delta u + k^2 u = 0$ in $\mathbb{R}^2$, which represents the vibration of a fixed membrane [14]. In a rectangular membrane the separation of variables leads to two Sturm–Liouville equations that can be solved independently. In a circular membrane the two equations (the angular and the radial) form a triangular system and cannot be solved independently. We can solve them one by one by inserting the parameter from the solution of the angular equation into the radial equation. In an elliptic membrane the separation leads to the Mathieu and modified Mathieu equations (see, e.g., [23])

$$\begin{aligned} y_1''(x_1) + (2\lambda \cosh 2x_1 - \mu) y_1(x_1) &= 0, \\ y_2''(x_2) - (2\lambda \cos 2x_2 - \mu) y_2(x_2) &= 0, \end{aligned} \tag{1.3}$$

which have to be solved simultaneously and thus form a genuine two-parameter eigenvalue problem.

Another problem that can be cast in the form (1.1) is the three-point boundary problem [7]. A typical problem is

$$-(p(x)y'(x))' + q(x)y(x) = (\lambda r(x) + \mu s(x))y(x), \tag{1.4}$$

subject to $y(a) = y(b) = y(c) = 0$, where $a < b < c$. We can treat (1.4) as a two-parameter eigenvalue problem

$$-(p(x_i)y_i'(x_i))' + q(x_i)y_i(x_i) = (\lambda r(x_i) + \mu s(x_i))y_i(x_i)$$

for $i = 1, 2$, where $x_1 \in [a, b]$, $x_2 \in [b, c]$, and the boundary conditions are $y_1(a) = y_1(b) = y_2(b) = y_2(c) = 0$. An example (see [23] for details) is Lamé's equation

$$y''(x) + \frac{1}{2}\left(\frac{1}{x-a} + \frac{1}{x-b} + \frac{1}{x-c}\right) y'(x) + \frac{\lambda + \mu x}{(x-a)(x-b)(x-c)} y(x) = 0,$$

which arises in the solution of Laplace's equation in elliptic coordinates.

Two-parameter problems appear in the algebraic form (1.1) as well. In [16], it is shown that the optimal value of the relaxation parameter ω in the method of successive overrelaxation for a separable elliptic partial differential equation in two independent variables can be obtained from the eigenvalues of a certain two-parameter eigenvalue problem. In [15], algorithms for the estimation of material electrical properties from measurements of interdigital dielectrometry sensors are discussed. When the sensors are applied to the material that is composed of two layers, the properties of the individual layers are the eigenvalues of the appropriate two-parameter eigenvalue problem. Yet another example is dynamic model updating [6]. Suppose that we have a spring-mass model where the mass matrix is known and the stiffness parameter values of two springs have to be updated based on outside measurements of the natural frequencies. The updated parameters are the eigenvalues of a two-parameter problem. The above examples show the need for numerical solvers of problem (1.1).

Two-parameter problems can be expressed as two coupled generalized eigenvalue problems as follows. On the tensor product space $S := \mathbb{C}^{n_1} \otimes \mathbb{C}^{n_2}$ of dimension $N := n_1 n_2$ we define

$$\begin{aligned} \Delta_0 &= B_1 \otimes C_2 - C_1 \otimes B_2, \\ \Delta_1 &= A_1 \otimes C_2 - C_1 \otimes A_2, \\ \Delta_2 &= B_1 \otimes A_2 - A_1 \otimes B_2. \end{aligned}$$

(For details on the tensor product and relation to the multiparameter eigenvalue problem, see, for example, [2].) We assume that the two-parameter problem (1.1) is *nonsingular*; that is, the corresponding operator determinant Δ_0 is invertible. In this case $\Gamma_1 := \Delta_0^{-1}\Delta_1$ and $\Gamma_2 := \Delta_0^{-1}\Delta_2$ commute, and problem (1.1) is equivalent to the associated problem

$$\begin{aligned} \Delta_1 z &= \lambda \Delta_0 z, \\ \Delta_2 z &= \mu \Delta_0 z \end{aligned} \tag{1.5}$$

for decomposable tensors $z \in S$, $z = x \otimes y$ (see [2]). The left and right eigenvectors of (1.1) are Δ_0-orthogonal; i.e., if $x_1 \otimes x_2$ and $y_1 \otimes y_2$ are right and left eigenvectors, respectively, of (1.1), corresponding to distinct eigenvalues, then

$$(y_1 \otimes y_2)^* \Delta_0 (x_1 \otimes x_2) = \begin{vmatrix} y_1^* B_1 x_1 & y_1^* C_1 x_1 \\ y_2^* B_2 x_2 & y_2^* C_2 x_2 \end{vmatrix} = 0.$$

If (λ, μ) is an eigenvalue of (1.1), then

$$\dim \left(\bigcap_{\substack{i_1+i_2=N \\ i_1, i_2 \geq 0}} \operatorname{Ker}[(\Gamma_1 - \lambda I)^{i_1} (\Gamma_2 - \mu I)^{i_2}] \right)$$

is the *algebraic multiplicity* of (λ, μ). We say that (λ, μ) is *algebraically simple* when its algebraic multiplicity is one.

The following lemma is a consequence of Lemma 3 in [13].

LEMMA 1.1. *If (λ, μ) is an algebraically simple eigenvalue of the two-parameter eigenvalue problem* (1.1) *and $x_1 \otimes x_2$ and $y_1 \otimes y_2$ are the corresponding right and left eigenvectors, respectively, then the matrix*

$$\begin{bmatrix} y_1^* B_1 x_1 & y_1^* C_1 x_1 \\ y_2^* B_2 x_2 & y_2^* C_2 x_2 \end{bmatrix}$$

is nonsingular.

There exist some numerical methods for two-parameter eigenvalue problems. Most of them require that the problem be real and *right definite*, i.e., that all matrices A_i, B_i, and C_i be real symmetric and that Δ_0 be positive definite, and as a consequence, eigenvalues and eigenvectors are real. Most of the presented two-parameter problems are right definite (for instance, (1.3) and the one in [16]), but not all (for instance, the one in [15] where the eigenvalues are complex). It is the aim of this paper to introduce an algorithm for such non–right definite two-parameter eigenvalue problems.

One of the algorithms (also usable for large sparse matrices) for the right definite two-parameter problem is a Jacobi–Davidson type method [10], and ideas from this method are generalized in this paper to handle all nonsingular two-parameter eigenvalue problems.

One possible approach for solving (1.1) is to solve the associated couple of generalized problems (1.5). In the right definite case this can be achieved by numerical methods for simultaneous diagonalization of commutative symmetric matrices [12, 20, 5], while an algorithm for the general nonsingular case using the QZ algorithm is presented in this paper (see Algorithm 2.3). Solving the problem via the

associated problem is only feasible for problems of low dimension as the size of the matrices of the associated problem is $N \times N$.

Another method that can be used for non–right definite two-parameter problems of moderate size is Newton's method [4], which has the deficiency that it requires initial approximations close enough to the solution in order to avoid misconvergence. The continuation method [18] can be used for *weakly elliptic* problems, i.e., such that A_i, B_i, and C_i are real symmetric and one of B_i, C_i is positive definite. We mention that right definite two-parameter problems are also weakly elliptic [17, Lemma 2.1].

In this paper, we introduce a new Jacobi–Davidson type method that can be used to compute selected eigenpairs for nonsingular problems. The method works even without close initial approximations and is suitable for large sparse matrices. Our method computes the eigenvalue (λ, μ) of (1.1), which is closest to a given target $(\lambda_{\mathrm{T}}, \mu_{\mathrm{T}}) \in \mathbb{C}^2$, i.e., the one with minimum $|\lambda - \lambda_{\mathrm{T}}|^2 + |\mu - \mu_{\mathrm{T}}|^2$.

The outline of the paper is as follows. In section 2, we present a new algorithm for the computation of eigenpairs using the associated problem. This method is only suitable for matrices of moderate size, so we combine it with a subspace method. We generalize the Petrov–Galerkin approach to two-parameter eigenvalue problems in section 3. In section 4, we present a two-sided Jacobi–Davidson type method for two-parameter eigenvalue problems. Several possible correction equations are discussed in section 5. In section 6, we present a selection technique that allows the computation of more than one eigenpair. The time complexity is given in section 7, and some numerical examples are presented in section 8. Conclusions are summarized in section 9.

2. Algorithm based on the associated problem. We propose the following method for solving (1.1) via the associated problem (1.5). First we compute a QZ decomposition (generalized Schur form; see, e.g., [8]) of the matrix pencil (Δ_1, Δ_0). We obtain unitary matrices Q and Z such that $Q^*\Delta_0 Z = R$ and $Q^*\Delta_1 Z = S$ are upper triangular. Since Δ_0 is nonsingular, the same is true for R. From

$$\Delta_0^{-1}\Delta_1 = ZR^{-1}SZ^*$$

it follows that the eigenvalues of the first generalized eigenvalue problem in (1.5) are the quotients s_{ii}/r_{ii} of the diagonal elements of matrices S and R.

Next, we sort the generalized Schur form so that multiple eigenvalues of the first generalized eigenvalue problem in (1.5) appear in blocks (see, for instance, [22]). Let us assume that the generalized Schur form is sorted to meet this requirement, and let matrix $R^{-1}S$ be partitioned accordingly as

$$R^{-1}S = \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1p} \\ 0 & L_{22} & \cdots & L_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L_{pp} \end{bmatrix}. \tag{2.1}$$

In the above partition, multiple eigenvalues of $\Delta_0^{-1}\Delta_1$ are clustered in upper triangular matrices $L_{11}, \ldots, L_{pp}$ along the diagonal so that $\lambda(L_{ii}) \neq \lambda(L_{jj})$ for $i \neq j$, where $\lambda(L_{kk})$ is the eigenvalue of a block L_{kk}. Let us denote the size of L_{ii} by m_i for $i = 1, \ldots, p$.

LEMMA 2.1. *Let*

$$L = \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1p} \\ 0 & L_{22} & \cdots & L_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L_{pp} \end{bmatrix}$$

be a partitioning of a block upper triangular matrix L such that $\Lambda(L_{11}), \ldots, \Lambda(L_{pp})$ are mutually disjoint, where $\Lambda(L_{kk})$ is the set of eigenvalues of L_{kk}. If M commutes with L, then M is block upper triangular partitioned conformally with L.

Proof. First we study the case $p = 2$. Let M be partitioned conformally with L as

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}.$$

From $LM - ML = 0$ and the above assumption we obtain the equation $L_{22}M_{21} - M_{21}L_{11} = 0$. Because L_{11} and L_{22} have no eigenvalues in common, this is a nonsingular homogeneous Sylvester equation for M_{21} (see, for example, [21, p. 223]). Therefore, the unique solution is $M_{21} = 0$.

In the case $p > 2$, one can see that M is block upper triangular by applying the above argument on all appropriate 2×2 block partitions of L and M. ∎

LEMMA 2.2. *$T = Q^*\Delta_2 Z$ partitioned conformally with* (2.1) *is block upper triangular.*

Proof. As $\Delta_0^{-1}\Delta_1$ and $\Delta_0^{-1}\Delta_2$ commute, so do $R^{-1}S$ and $R^{-1}T$. It follows from Lemma 2.1 that $R^{-1}T$ is block upper triangular partitioned conformally to (2.1). As block upper triangular matrices keep their shape when multiplied by a triangular matrix, it follows from $T = R(R^{-1}T)$ that T is block upper triangular as well. ∎

Once R, S, and T are partitioned conformally with (2.1) as

$$R = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1p} \\ 0 & R_{22} & \cdots & R_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{pp} \end{bmatrix}, \qquad S = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1p} \\ 0 & S_{22} & \cdots & S_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S_{pp} \end{bmatrix},$$

$$T = \begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1p} \\ 0 & T_{22} & \cdots & T_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_{pp} \end{bmatrix},$$

it is straightforward to compute eigenvalues of (1.1). To each diagonal block L_{ii} of size m_i in $R^{-1}S$ correspond m_i eigenvalues $(\lambda_i, \mu_{i1}), \ldots, (\lambda_i, \mu_{im_i})$, where λ_i is the eigenvalue of L_{ii} and $\mu_{i1}, \ldots, \mu_{im_i}$ are eigenvalues of the generalized eigenvalue problem $T_{ii}w = \mu R_{ii}w$.

Now that we have all eigenvalues (λ_j, μ_j), $j = 1, \ldots, N$, of (1.1), we compute the corresponding eigenvectors $x_{j1} \otimes x_{j2}$. We do this by solving $(A_i - \lambda_j B_i - \mu_j C_i)x_{ji} = 0$, where x_{ji} is normalized, for $i = 1, 2$. In a similar way we can obtain left eigenvectors $y_{j1} \otimes y_{j2}$ when they are required.

The complete procedure is summarized in Algorithm 2.3.

ALGORITHM 2.3. An algorithm for the nonsingular two-parameter eigenvalue problem (1.1).

1. Compute Δ_0, Δ_1, and Δ_2 of the associated problem (1.5).
2. Compute a generalized Schur decomposition $Q^*\Delta_0 Z = R$ and $Q^*\Delta_1 Z = S$, such that Q and Z are unitary, R and S are upper triangular, and the Schur form is sorted so that multiple values of $\lambda_i := s_{ii}/r_{ii}$ are clustered along the diagonal of $R^{-1}S$. As a result of this, R and S are partitioned as

$$R = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1p} \\ 0 & R_{22} & \cdots & R_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{pp} \end{bmatrix}, \qquad S = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1p} \\ 0 & S_{22} & \cdots & S_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S_{pp} \end{bmatrix},$$

   where the size of R_{ii} and S_{ii} is m_i and $m_1 + \cdots + m_p = N$.
3. Compute diagonal blocks $T_{11}, \ldots, T_{pp}$ of $T = Q^*\Delta_2 Z$, partitioned conformally with R and S.
4. Compute the eigenvalues $\mu_{i1}, \ldots, \mu_{im_i}$ of the generalized eigenvalue problem

$$T_{ii}w = \mu R_{ii}w$$

   for $i = 1, \ldots, p$.
5. The eigenvalues of (1.1) are

$$(\lambda_1, \mu_{11}), \ldots, (\lambda_1, \mu_{1m_1}); \ldots ; (\lambda_p, \mu_{p1}), \ldots, (\lambda_p, \mu_{pm_p});$$

   reindex them as $(\lambda_1, \mu_1), \ldots, (\lambda_N, \mu_N)$.
6. For each eigenvalue (λ_j, μ_j), $j = 1, \ldots, N$, of (1.1), take for x_{ji} and y_{ji} the smallest right and the smallest left singular vector of $A_i - \lambda_j B_i - \mu_j C_i$, respectively, for $i = 1, 2$ (see Remark 2.5).

Remark 2.4. In numerical computation we may cluster not only multiple eigenvalues but also close eigenvalues of $R^{-1}S$. After clustering, we take the mean of all eigenvalues in the cluster of size m_i as a multiple eigenvalue of order m_i. This means that we take λ_i as a mean of all eigenvalues of the generalized eigenvalue problem

$$S_{ii}w = \lambda R_{ii}w$$

for $i = 1, \ldots, p$.

Remark 2.5. In practice there will be an error in a detected eigenvalue (λ_j, μ_j). Therefore we take the right singular vector corresponding to the smallest singular value to find the normalized x_{ji} such that $(A_i - \lambda_j B_i - \mu_j C_i)x_{ji} \approx 0$ for $i = 1, 2$. In a similar way we get the approximation to the left eigenvector.

Let us assume that A_i, B_i, C_i are dense and that $n_1 = n_2 = n$. The time complexity of Algorithm 2.3 is $\mathcal{O}(n^6)$ for the computation of eigenvalues using QZ decomposition of matrices of size n^2. The maximum additional work for eigenvectors is $\mathcal{O}(n^5)$, as we have to compute $\mathcal{O}(n^2)$ singular value decompositions of matrices of size n. If we are not interested in all eigenvectors (as is often the case for large sparse matrices), then the additional work can be substantially less.

The large time complexity is the reason that Algorithm 2.3 is useful only for matrices of a modest size. For larger problems we embed this method in a subspace method and use Algorithm 2.3 for the small projected problems.

3. Subspace methods and Petrov triples. In this section we study subspace methods for the two-parameter eigenvalue problem. In a subspace method we start

with a given search subspace, from which approximations to eigenpairs are computed (*extraction*). In the extraction we usually have to solve a smaller eigenvalue problem of the same type as the original one. After each step we expand the subspace by a new direction (*expansion*), and as the search subspace grows, the eigenpair approximations will in general converge to an eigenpair of the original problem. In this section we discuss the extraction, and in the next section we discuss the algorithm and the expansion.

Suppose that we have k-dimensional search spaces $\mathcal{U}_{ik} \subset \mathbb{C}^{n_i}$ and k-dimensional test spaces $\mathcal{V}_{ik} \subset \mathbb{C}^{n_i}$ for $i = 1, 2$. Let the columns of the $n_i \times k$ matrices U_{ik} and V_{ik} form orthogonal bases for $\mathcal{U}_{ik}$ and $\mathcal{V}_{ik}$, respectively, for $i = 1, 2$. The Petrov–Galerkin conditions

$$\begin{aligned}(A_1 - \sigma B_1 - \tau C_1)u_1 &\perp \mathcal{V}_{1k},\\ (A_2 - \sigma B_2 - \tau C_2)u_2 &\perp \mathcal{V}_{2k},\end{aligned}$$

where $u_i \in \mathcal{U}_{ik}\backslash\{0\}$ for $i = 1, 2$, lead to the smaller projected two-parameter problem

$$\begin{aligned}V_{1k}^* A_1 U_{1k} c_1 &= \sigma V_{1k}^* B_1 U_{1k} c_1 + \tau V_{1k}^* C_1 U_{1k} c_1,\\ V_{2k}^* A_2 U_{2k} c_2 &= \sigma V_{2k}^* B_2 U_{2k} c_2 + \tau V_{2k}^* C_2 U_{2k} c_2,\end{aligned} \tag{3.1}$$

where $u_i = U_{ik} c_i \neq 0$ for $i = 1, 2$ and $\sigma, \tau \in \mathbb{C}$.

We say that an eigenvalue (σ, τ) of (3.1) is a *Petrov value* for the two-parameter eigenvalue problem (1.1) with respect to the search spaces $\mathcal{U}_{1k}$ and $\mathcal{U}_{2k}$ and test spaces $\mathcal{V}_{1k}$ and $\mathcal{V}_{2k}$. If (σ, τ) is an eigenvalue of (3.1) and $c_1 \otimes c_2$ is the corresponding right eigenvector, then $u_1 \otimes u_2$ is a *right Petrov vector*. Similarly, if $d_1 \otimes d_2$ is the corresponding left eigenvector of (3.1), then $v_1 \otimes v_2$ is a *left Petrov vector*, where $v_i = V_{ik} d_i$ for $i = 1, 2$. It is easy to check that σ and τ are equal to the *two-sided tensor Rayleigh quotients*

$$\begin{aligned}\sigma = \rho_1(u, v) &= \frac{(v_1 \otimes v_2)^* \Delta_1 (u_1 \otimes u_2)}{(v_1 \otimes v_2)^* \Delta_0 (u_1 \otimes u_2)} = \frac{(v_1^* A_1 u_1)(v_2^* C_2 u_2) - (v_1^* C_1 u_1)(v_2^* A_2 u_2)}{(v_1^* B_1 u_1)(v_2^* C_2 u_2) - (v_1^* C_1 u_1)(v_2^* B_2 u_2)},\\ \tau = \rho_2(u, v) &= \frac{(v_1 \otimes v_2)^* \Delta_2 (u_1 \otimes u_2)}{(v_1 \otimes v_2)^* \Delta_0 (u_1 \otimes u_2)} = \frac{(v_1^* B_1 u_1)(v_2^* A_2 u_2) - (v_1^* A_1 u_1)(v_2^* B_2 u_2)}{(v_1^* B_1 u_1)(v_2^* C_2 u_2) - (v_1^* C_1 u_1)(v_2^* B_2 u_2)}.\end{aligned} \tag{3.2}$$

In order to obtain Petrov values, we have to solve small two-parameter eigenvalue problems. For this purpose, we use Algorithm 2.3. Altogether, we obtain k^2 *Petrov triples* $((\sigma_j, \tau_j), u_{j1} \otimes u_{j2}, v_{j1} \otimes v_{j2})$ that are approximations to eigentriples $((\lambda_j, \mu_j), x_{j1} \otimes x_{j2}, y_{j1} \otimes y_{j2})$ of (1.1) for $j = 1, \ldots, k^2$.

4. Jacobi–Davidson type method. The Jacobi–Davidson method [19] is one of the subspace methods that may be used for the numerical solution of one-parameter eigenvalue problems. For an overview of subspace methods, see, for example, [3]. In the Jacobi–Davidson method approximate solutions to certain correction equations are used to expand the search space. The search for a new direction is restricted to the subspace that is orthogonal or oblique to the last chosen right (or left) Petrov vector.

A Jacobi–Davidson type method has been successfully applied to the right definite two-parameter eigenvalue problem [10]. The method in [10] is one-sided, which means that the search spaces $\mathcal{V}_i$ in (3.1) are the same as the test spaces $\mathcal{U}_i$. When we tested the one-sided method from [10] on non–right definite problems, it turned out

that the performance was sometimes not optimal; in particular, there were problems with convergence to unwanted eigenvalues or no convergence at all. Therefore we generalize the two-sided Jacobi–Davidson method [9] to two-parameter eigenvalue problems. The idea is to take $\mathcal{U}_i$ as search spaces for the right eigenvectors, and $\mathcal{V}_i$ as search spaces for the left eigenvectors. An advantage of a two-sided method is that both the left and the right eigenvectors are approximated, which implies an accurate approximation of the eigenvalue (see Lemma 5.1). An obvious disadvantage is that such an approach requires more memory and twice the work (in terms of matrix-vector multiplications) for one iteration. Numerical experiments in section 8 indicate that for non–right definite problems the two-sided Jacobi–Davidson type method often gives better results than the one-sided method.

A sketch of the two-sided Jacobi–Davidson type method for the two-parameter problem is presented in Algorithm 4.1. In step 2(b) we have to choose a Petrov triple. Some options are given later in this section. In step 2(e), we have to find new search directions in order to expand the search and test subspaces. We discuss several possible correction equations in section 5.

ALGORITHM 4.1. A two-sided Jacobi–Davidson type method for the nonsingular two-parameter eigenvalue problem.

1. **Start.** Choose initial vectors u_1, u_2, v_1, and v_2 with unit norm.
   (a) Set $U_{i1} = [u_i]$, $V_{i1} = [v_i]$ for $i = 1, 2$.
   (b) Set $k = 1$.
2. **Iterate.** Until convergence or $k > k_{\max}$ do:
   (a) Solve the projected two-parameter eigenvalue problem (3.1) by Algorithm 2.3.
   (b) Select an appropriate Petrov value (σ, τ) and the corresponding right and left Petrov vectors $u_1 \otimes u_2$ and $v_1 \otimes v_2$, where $u_i = U_{ik}c_i$, $v_i = V_{ik}d_i$ for $i = 1, 2$, respectively.
   (c) Compute the right and left residuals

$$r_i^R = (A_i - \sigma B_i - \tau C_i)u_i, \tag{4.1}$$

$$r_i^L = (A_i - \sigma B_i - \tau C_i)^* v_i \tag{4.2}$$

   for $i = 1, 2$.
   (d) Stop if $\rho_k \leq \varepsilon$, where

$$\rho_k = (\|r_1^R\|^2 + \|r_2^R\|^2 + \|r_1^L\|^2 + \|r_2^L\|^2)^{1/2}. \tag{4.3}$$

   (e) Solve approximately one of the proposed correction equations (see section 5), and obtain new directions s_i and t_i for $i = 1, 2$.
   (f) Expand the search subspaces. Set

$$U_{i,k+1} = \mathrm{RGS}(U_{ik}, s_i),$$
$$V_{i,k+1} = \mathrm{RGS}(V_{ik}, t_i),$$

   where RGS denotes the repeated Gram–Schmidt orthonormalization, for $i = 1, 2$.
   (g) Set $k = k + 1$.
   (h) Restart. If the dimension of the image of U_{ik} and V_{ik} exceeds $l_{\max}$, then replace U_{ik}, V_{ik} with new orthonormal bases of dimension $l_{\min}$.

To apply this algorithm, we need to specify a target $(\lambda_{\mathrm{T}}, \mu_{\mathrm{T}})$, a tolerance ε, a maximum number of steps $k_{\max}$, a maximum dimension of the search subspaces $l_{\max}$,

and a number $l_{\min} < l_{\max}$ that specifies the dimension of the search subspaces after a restart. As Algorithm 2.3 is able to solve only low-dimensional two-parameter problems (3.1) in a reasonable time, we expand the search spaces up to the preselected dimension $l_{\max}$ and then restart the algorithm. For a restart, we take the $l_{\min}$ eigenvector approximations with the smallest residuals (4.3) as a basis for the initial search space.

We also have to specify a criterion for step 2(b). Suppose that we are looking for the eigenvalue closest to the target $(\lambda_{\mathrm{T}}, \mu_{\mathrm{T}})$. We suggest combining two approaches. In the first part we select the Petrov value (σ, τ) closest to the target until the residual ρ_k drops below $\varepsilon_{\text{change}}$. In the second part we take the Petrov triple with the smallest residual (4.3). Both stages can be seen as an accelerated inexact Rayleigh quotient iteration.

Remark 4.2. In step 2(d) we could also stop the algorithm if either the norm of the right residuals r_1^R and r_2^R or the norm of the left residuals r_1^L and r_2^L is small enough. In either case we can expect that (σ, τ) is a good approximation to an eigenvalue, and we can compute the corresponding right or left eigenvectors by solving one (orthogonal) correction equation; see also [9].

In the following section we discuss the expansion in step 2(e) and derive several correction equations.

5. Correction equations. Let (σ, τ) be a Petrov value that approximates the eigenvalue (λ, μ) of (1.1), and let $u_1 \otimes u_2$ and $v_1 \otimes v_2$ be its corresponding left and right Petrov vectors, respectively. Let us assume that u_1, u_2, v_1, and v_2 are normalized.

We are searching for orthogonal improvements of the left and right Petrov vectors of the form

$$(A_i - \lambda B_i - \mu C_i)(u_i + s_i) = 0, \tag{5.1}$$

$$(A_i - \lambda B_i - \mu C_i)^*(v_i + t_i) = 0, \tag{5.2}$$

where $s_i \perp a_i$ and $t_i \perp b_i$ for $i = 1, 2$. We will discuss the choices for a_i and b_i later; at this time we require just that $a_i \not\perp u_i$ and $b_i \not\perp v_i$.

Using (4.1) and (4.2), we can rewrite (5.1) and (5.2) as

$$\begin{aligned}(A_i - \sigma B_i - \tau C_i)s_i = {} & -r_i^R + (\lambda - \sigma)B_i u_i + (\mu - \tau)C_i u_i \\ & + (\lambda - \sigma)B_i s_i + (\mu - \tau)C_i s_i,\end{aligned} \tag{5.3}$$

$$\begin{aligned}(A_i - \sigma B_i - \tau C_i)^* t_i = {} & -r_i^L + (\lambda - \sigma)^* B_i^* v_i + (\mu - \tau)^* C_i^* v_i \\ & + (\lambda - \sigma)^* B_i^* t_i + (\mu - \tau)^* C_i^* t_i.\end{aligned} \tag{5.4}$$

LEMMA 5.1. *If $u_i = x_i - s_i$ and $v_i = y_i - t_i$, for $i = 1, 2$, are close enough approximations to a left and a right eigenvector of* (1.1) *for the same algebraically simple eigenvalue (λ, μ), then the two-sided Rayleigh quotient $(\sigma, \tau) = (\rho_1(u, v), \rho_2(u, v))$ is an $\mathcal{O}(\|s_1\|\|t_1\| + \|s_2\|\|t_2\|)$ approximation to (λ, μ); i.e.,*

$$\left\| \begin{bmatrix} \lambda - \sigma \\ \mu - \tau \end{bmatrix} \right\| = \mathcal{O}(\|s_1\|\|t_1\| + \|s_2\|\|t_2\|). \tag{5.5}$$

Proof. We write the residual (4.1) as

$$r_i^R = -(A_i - \lambda B_i - \mu C_i)s_i + (\lambda - \sigma)B_i u_i + (\mu - \tau)C_i u_i. \tag{5.6}$$

When we multiply (5.6) by v_i^* and take into account that $v_i^* r_i^R = 0$ and

$$v_i^*(A_i - \lambda B_i - \mu C_i) = -t_i^*(A_i - \lambda B_i - \mu C_i)$$

for $i = 1, 2$, then we obtain

$$\begin{bmatrix} v_1^* B_1 u_1 & v_1^* C_1 u_1 \\ v_2^* B_2 u_2 & v_2^* C_2 u_2 \end{bmatrix} \begin{bmatrix} \lambda - \sigma \\ \mu - \tau \end{bmatrix} = - \begin{bmatrix} t_1^*(A_1 - \lambda B_1 - \mu C_1) s_1 \\ t_2^*(A_2 - \lambda B_2 - \mu C_2) s_2 \end{bmatrix}. \tag{5.7}$$

If $\|s_i\|$ and $\|t_i\|$ are small enough, then (5.7) is a nonsingular system because of Lemma 1.1 and continuity. We can deduce from (5.7) that

$$\left\| \begin{bmatrix} \lambda - \sigma \\ \mu - \tau \end{bmatrix} \right\| = \left\| \begin{bmatrix} v_1^* B_1 u_1 & v_1^* C_1 u_1 \\ v_2^* B_2 u_2 & v_2^* C_2 u_2 \end{bmatrix}^{-1} \begin{bmatrix} t_i^*(A_1 - \lambda B_1 - \mu C_1) s_i \\ t_2^*(A_2 - \lambda B_2 - \mu C_2) s_i \end{bmatrix} \right\|$$

and so obtain (5.5). ∎

It follows from Lemma 5.1 that asymptotically (i.e., when we have good approximate right and left eigenvectors) we can consider s_i and t_i as first-order corrections and $(\lambda - \sigma)B_i u_i + (\mu - \tau)C_i u_i$ and $(\lambda - \sigma)^* B_i^* v_i + (\mu - \tau)^* C_i^* v_i$ as second-order corrections, and finally, $(\lambda - \sigma)B_i s_i + (\mu - \tau)C_i s_i$ and $(\lambda - \sigma)^* B_i^* t_i + (\mu - \tau)^* C_i^* t_i$ can be interpreted as third-order corrections.

5.1. First-order–based correction equations. If we ignore the second- and higher-order terms in (5.3), then we obtain the equation

$$(A_i - \sigma B_i - \tau C_i) s_i = -r_i^R. \tag{5.8}$$

Because r_i^R is orthogonal to v_i, we can multiply (5.8) with an oblique projection $I - \frac{c_i v_i^*}{v_i^* c_i}$, where $c_i \not\perp v_i$, that does not change r_i^R. Secondly, since s_i is orthogonal to a_i, we can write $(I - \frac{u_i a_i^*}{a_i^* u_i}) s_i$ instead of s_i. Thus we obtain the correction equation for the vector u_i,

$$\left(I - \frac{c_i v_i^*}{v_i^* c_i} \right) (A_i - \sigma B_i - \tau C_i) \left(I - \frac{u_i a_i^*}{a_i^* u_i} \right) s_i = -r_i^R \tag{5.9}$$

for $i = 1, 2$. In a similar way we obtain from (5.4) the correction equation for the vector v_i,

$$\left(I - \frac{d_i u_i^*}{u_i^* d_i} \right) (A_i - \sigma B_i - \tau C_i)^* \left(I - \frac{v_i b_i^*}{b_i^* v_i} \right) t_i = -r_i^L \tag{5.10}$$

for $i = 1, 2$, where $d_i \not\perp u_i$.

We solve these correction equations only approximately, using, for instance, some Krylov subspace method. Since the operator in (5.9) maps $a_i^\perp$ onto $v_i^\perp$, it is suitable to take $a_i = v_i$ in order to apply the Krylov solver without a preconditioner (see, for example, the discussion in [9, section 4.2]). If $a_i \neq v_i$, then we need a preconditioner that maps the image space $v_i^\perp$ bijectively onto $a_i^\perp$. Similarly, we need a preconditioner for (5.10) when $b_i \neq u_i$.

Different choices of vectors a_i, b_i, c_i, d_i lead to different correction equations. We discuss some options.

1. For the first correction equation we take $a_i = d_i = v_i$, $b_i = c_i = u_i$. We obtain a pair of correction equations

$$\begin{aligned}\left(I - \frac{u_i v_i^*}{v_i^* u_i}\right)(A_i - \sigma B_i - \tau C_i)\left(I - \frac{u_i v_i^*}{v_i^* u_i}\right) s_i &= -r_i^R,\\ \left(I - \frac{v_i u_i^*}{u_i^* v_i}\right)(A_i - \sigma B_i - \tau C_i)^*\left(I - \frac{v_i u_i^*}{u_i^* v_i}\right) t_i &= -r_i^L\end{aligned} \tag{5.11}$$

for $s_i \perp v_i$, $t_i \perp u_i$ for $i = 1, 2$. The operator in the first equation is the conjugate transpose of the operator in the second equation, and we can solve these equations simultaneously by biconjugate gradients (BiCG). It is also possible to solve equations in (5.11) separately by the generalized minimum residual method (GMRES).

2. For this correction equation we take $a_i = c_i = u_i$, $b_i = d_i = v_i$.

It is a natural approach for (5.9) and (5.10) to take $a_i = u_i$ and $b_i = v_i$, as in this case we are looking for updates orthogonal to the current approximation. As it turns out later in section 5.2, when we use preconditioning, an interesting choice for c_i and d_i is to take $c_i = u_i$ and $d_i = v_i$, which leads to a pair of correction equations

$$\begin{aligned}\left(I - \frac{u_i v_i^*}{v_i^* u_i}\right)(A_i - \sigma B_i - \tau C_i)(I - u_i u_i^*) s_i &= -r_i^R,\\ \left(I - \frac{v_i u_i^*}{u_i^* v_i}\right)(A_i - \sigma B_i - \tau C_i)^*(I - v_i v_i^*) t_i &= -r_i^L\end{aligned} \tag{5.12}$$

for $s_i \perp u_i$, $t_i \perp v_i$ for $i = 1, 2$. In order to solve (5.12) approximately by a Krylov solver we need a preconditioner because $a_i \neq v_i$; see section 5.2.

3. In this case we take $a_i = u_i$, $b_i = v_i$, $c_i = g_i$, $d_i = h_i$, where

$$\begin{aligned} g_i &= (\lambda_{\mathrm{T}} - \sigma) B_i u_i + (\mu_{\mathrm{T}} - \tau) C_i u_i,\\ h_i &= (\lambda_{\mathrm{T}} - \sigma)^* B_i^* v_i + (\mu_{\mathrm{T}} - \tau)^* C_i^* v_i.\end{aligned}$$

The idea behind the choice of c_i and d_i is that when the target $(\lambda_{\mathrm{T}}, \mu_{\mathrm{T}})$ is close to the eigenvalue, then the projections with g_i and h_i almost annihilate the second-order terms in (5.3) and (5.4) and thus reduce the neglected quantity.

We derive the correction equations

$$\begin{aligned}\left(I - \frac{g_i v_i^*}{v_i^* g_i}\right)(A_i - \sigma B_i - \tau C_i)(I - u_i u_i^*) s_i &= -r_i^R,\\ \left(I - \frac{h_i u_i^*}{u_i^* h_i}\right)(A_i - \sigma B_i - \tau C_i)^*(I - v_i v_i^*) t_i &= -r_i^L\end{aligned} \tag{5.13}$$

for $s_i \perp u_i$, $t_i \perp v_i$ for $i = 1, 2$. Again, if we want to solve (5.13) approximately by a Krylov solver, then we need a preconditioner, as $a_i \neq v_i$; see the next section.

5.2. Preconditioned first-order–based correction equations. We have mentioned that we need a preconditioner for a Krylov solver for the correction equation (5.9) when the domain subspace $a_i^\perp$ and the range subspace $v_i^\perp$ do not agree. However, we can also use a preconditioner when the domain and the range agree, to speed up the convergence.

Suppose that a left preconditioner M_i is available for $A_i - \sigma B_i - \tau C_i$ such that $M_i^{-1}(A_i - \sigma B_i - \tau C_i) \approx I$. A calculation shows that if we assume that $a_i^* M_i^{-1} c_i \neq 0$,

then the inverse of the map

$$\left(I - \frac{c_i v_i^*}{v_i^* c_i}\right) M_i \left(I - \frac{u_i a_i^*}{a_i^* u_i}\right)$$

from $a_i^\perp$ to $v_i^\perp$ is the map

$$\left(I - \frac{M_i^{-1} c_i a_i^*}{a_i^* M_i^{-1} c_i}\right) M_i^{-1} \left(I - \frac{c_i v_i^*}{v_i^* c_i}\right)$$

from $v_i^\perp$ to $a_i^\perp$. Therefore, using left preconditioning changes (5.9) into

$$\begin{aligned}\left(I - \frac{M_i^{-1} c_i a_i^*}{a_i^* M_i^{-1} c_i}\right) M_i^{-1} \left(I - \frac{c_i v_i^*}{v_i^* c_i}\right) (A_i - \sigma B_i - \tau C_i) \left(I - \frac{u_i a_i^*}{a_i^* u_i}\right) s_i \\ = - \left(I - \frac{M_i^{-1} c_i a_i^*}{a_i^* M_i^{-1} c_i}\right) M_i^{-1} r_i^R\end{aligned}$$

for $i = 1, 2$. Correction equation (5.10) for the left eigenvector can be dealt with similarly. A preconditioner for $A_i - \sigma B_i - \tau C_i$ automatically suggests a preconditioner for $(A_i - \sigma B_i - \tau C_i)^*$.

We can combine different preconditioners with different correction equations. Here are some possibilities.

1. Our suggestion for the preconditioner is

$$M_i = A_i - \lambda_{\mathrm{T}} B_i - \mu_{\mathrm{T}} C_i, \tag{5.14}$$

where $(\lambda_{\mathrm{T}}, \mu_{\mathrm{T}})$ is the target. Instead of exact inversion we can also take an inexact inverse, for example one obtained using an incomplete LU decomposition.

2. The simplest option is to take the identity as a preconditioner in order to be able to use a Krylov solver for the correction equation. For example, if we take correction equation (5.12) and the identity as a preconditioner, then we have to multiply (5.9) and (5.10) by orthogonal projectors $I - u_i u_i^*$ and $I - v_i v_i^*$, respectively. From $(I - u_i u_i^*)(I - \frac{u_i v_i^*}{v_i^* u_i}) = I - u_i u_i^*$ and $(I - v_i v_i^*)(I - \frac{v_i u_i^*}{u_i^* v_i}) = I - v_i v_i^*$ we get

$$\begin{aligned}(I - u_i u_i^*)(A_i - \sigma B_i - \tau C_i)(I - u_i u_i^*) s_i &= -(I - u_i u_i^*) r_i^R, \\ (I - v_i v_i^*)(A_i - \sigma B_i - \tau C_i)^*(I - v_i v_i^*) t_i &= -(I - v_i v_i^*) r_i^L\end{aligned} \tag{5.15}$$

for $i = 1, 2$. One can recognize (5.15) as the correction equations of the standard Jacobi–Davidson method applied to $A_i - \sigma B_i - \tau C_i$ and $(A_i - \sigma B_i - \tau C_i)^*$.

5.3. Second-order–based correction equation. For this case we generalize the correction equation with oblique projections for the right definite two-parameter eigenvalue problem [10]. If we define

$$K = \begin{bmatrix} A_1 - \sigma B_1 - \tau C_1 & 0 \\ 0 & A_2 - \sigma B_2 - \tau C_2 \end{bmatrix},$$

$$r^R = \begin{bmatrix} r_1^R \\ r_2^R \end{bmatrix}, \qquad r^L = \begin{bmatrix} r_1^L \\ r_2^L \end{bmatrix},$$

then we can reformulate (5.3) and (5.4) (neglecting third-order correction terms) as

$$K \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = -r^R + (\lambda - \sigma) \begin{bmatrix} B_1 u_1 \\ B_2 u_2 \end{bmatrix} + (\mu - \tau) \begin{bmatrix} C_1 u_1 \\ C_2 u_2 \end{bmatrix} \tag{5.16}$$

and

$$K^* \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = -r^L + (\lambda - \sigma)^* \begin{bmatrix} B_1^* v_1 \\ B_2^* v_2 \end{bmatrix} + (\mu - \tau)^* \begin{bmatrix} C_1^* v_1 \\ C_2^* v_2 \end{bmatrix}. \tag{5.17}$$

Let V_{R} be a $(n_1 + n_2) \times 2$ matrix with orthonormal columns such that

$$\operatorname{span}(V_{\mathrm{R}}) = \operatorname{span}\left(\begin{bmatrix} B_1 u_1 \\ B_2 u_2 \end{bmatrix}, \begin{bmatrix} C_1 u_1 \\ C_2 u_2 \end{bmatrix} \right),$$

and let

$$W_{\mathrm{R}} = \begin{bmatrix} v_1 & 0 \\ 0 & v_2 \end{bmatrix}.$$

With the oblique projection

$$P_{\mathrm{R}} = I - V_{\mathrm{R}} (W_{\mathrm{R}}^* V_{\mathrm{R}})^{-1} W_{\mathrm{R}}^*$$

onto $\operatorname{span}(W_{\mathrm{R}})^{\perp}$ along $\operatorname{span}(V_{\mathrm{R}})$, it follows that

$$P_{\mathrm{R}} r^R = r^R \quad \text{and} \quad P_{\mathrm{R}} \begin{bmatrix} B_1 u_1 \\ B_2 u_2 \end{bmatrix} = P_{\mathrm{R}} \begin{bmatrix} C_1 u_1 \\ C_2 u_2 \end{bmatrix} = 0.$$

Therefore, from multiplying (5.16) by P_{R}, we obtain

$$P_{\mathrm{R}} K \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = -r^R.$$

Suppose that we are looking for corrections such that $s_i \perp v_i$ and $t_i \perp u_i$. Then

$$P_{\mathrm{R}} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix},$$

and the result is the correction equation

$$P_{\mathrm{R}} K P_{\mathrm{R}} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = -r^R. \tag{5.18}$$

Remark 5.2. If $u_1 \otimes u_2$ and $v_1 \otimes v_2$ are close approximations to eigenvectors $x_1 \otimes x_2$ and $y_1 \otimes y_2$, corresponding to a single eigenvalue of (1.1), then it follows from Lemma 1.1 that $W_R^* V_R$ is nonsingular. If the above is not true, then it is possible that V_R does not exist or that $W_R^* V_R$ is singular. In either of these two cases we can use one of the correction equations from section 5.1 to expand the search and test spaces.

In a similar manner we obtain a correction equation for t_1 and t_2. If V_{L}, W_{L}, and P_{L} are defined similarly for (5.17), then we have

$$P_{\mathrm{L}} K^* P_{\mathrm{L}} \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = -r^L. \tag{5.19}$$

We separately solve (5.18) and (5.19) approximately using a few steps of GMRES.

Better results can be expected if we use preconditioners. Suppose that M is a left preconditioner for K, for instance, a block preconditioner with the preconditioners M_i in (5.14) as blocks. One can show that if $W_{\mathrm{R}}^* M^{-1} V_{\mathrm{R}}$ is nonsingular, then the inverse of a map $P_{\mathrm{R}} M P_{\mathrm{R}}$ from $\mathrm{span}(W_{\mathrm{R}})^\perp$ to $\mathrm{span}(W_{\mathrm{R}})^\perp$ is

$$(I - M^{-1}V_{\mathrm{R}}(W_{\mathrm{R}}^* M^{-1} V_{\mathrm{R}})^{-1} W_{\mathrm{R}}^*) M^{-1} P_{\mathrm{R}}.$$

Thus we obtain a preconditioned correction equation

$$\begin{aligned}(I - M^{-1}V_{\mathrm{R}}(W_{\mathrm{R}}^* M^{-1} V_{\mathrm{R}})^{-1} W_{\mathrm{R}}^*) M^{-1} P_{\mathrm{R}} K P_{\mathrm{R}} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \\ = (I - M^{-1}V_{\mathrm{R}}(W_{\mathrm{R}}^* M^{-1} V_{\mathrm{R}})^{-1} W_{\mathrm{R}}^*) M^{-1} r^R.\end{aligned} \tag{5.20}$$

In a similar manner we get a preconditioned equation for t_1 and t_2.

5.4. One-sided approach. Instead of the two-sided, we could also apply the one-sided approach, where the search subspace is the same as the test subspace [10]. One-sided versions can be easily derived from the above two-sided correction equations. All one has to do is use $V_i = U_i$ for $i = 1, 2$, and solve only the correction equations for s_1 and s_2.

The advantage of the one-sided approach is that it requires less memory and roughly half the work for one outer iteration. On the other side, numerical results in section 8 show that the two-sided approach gives more accurate results. Also, if we use the one-sided approach, then we cannot apply Lemma 1.1 as we did in Remark 5.2.

6. Computing more eigenpairs. Suppose that we are interested in $p > 1$ eigenpairs of (1.1). In one-parameter eigenvalue problems various deflation techniques can be applied in order to compute more than one eigenpair. The difficulties that are met when we try to translate standard deflation ideas from one-parameter problems to two-parameter problems are discussed in [10].

For a general two-parameter eigenvalue problem we can apply a technique similar to that in [10] for the right definite problem using the Δ_0-orthogonality of left and right eigenvectors. Suppose that we have already found p eigenvalues (λ_i, μ_i) with the corresponding left and right eigenvectors $x_{1i} \otimes x_{2i}$ and $y_{1i} \otimes y_{2i}$ for $i = 1, \ldots, p$. Now we adjust Algorithm 4.1 so that in step 2(b) we consider only those Petrov triples for which $u_1 \otimes u_2$ and $v_1 \otimes v_2$ satisfy

$$\min(|(v_1 \otimes v_2)^* \Delta_0 (x_{1i} \otimes x_{2i})|, |(y_{1i} \otimes y_{2i})^* \Delta_0 (u_1 \otimes u_2)|) < \eta \quad \text{for } i = 1, \ldots, p \tag{6.1}$$

for an $\eta > 0$. A suggestion for η (used in Example 8.4 in section 8) is

$$\eta = \frac{1}{2} \min_{i=1,\ldots,p} ((y_{1i} \otimes y_{2i})^* \Delta_0 (x_{1i} \otimes x_{2i})).$$

If no triple satisfies this condition, then we take the one that gives the smallest left-hand side of (6.1).

Let us mention that an efficient way to compute (6.1) is to apply the relation (cf. (3.2))

$$(x_1 \otimes x_2)^* \Delta_0 (y_1 \otimes y_2) = (x_1^* B_1 y_1)(x_2^* C_2 y_2) - (x_1^* C_1 y_1)(x_2^* B_2 y_2).$$

If we want to compute more eigenpairs using the one-sided approach, then we have to compute the left eigenvectors separately for each converged eigenvalue. If we use the two-sided approach, then left and right eigenvectors are already computed.

7. Time complexity. The analysis of time complexity of Algorithm 4.1 is similar to the analysis for the Jacobi–Davidson algorithm for right definite two-parameter eigenvalue problems in [10, section 6]. Therefore, the details are omitted and the main results are stated.

If we assume that $n = n_1 = n_2$ and that m steps of GMRES are used for the approximate solutions of the correction equations, then the time complexity of one outer step of Algorithm 4.1 for dense matrices is $\mathcal{O}(mn^2)$. Also important is the storage requirement. If an algorithm works with matrices A_i, B_i, and C_i as Algorithm 4.1 does, then it requires $\mathcal{O}(n^2)$ memory. On the other hand, Algorithm 2.3, which works with the associated system (1.5), needs $\mathcal{O}(n^4)$ memory, which may fast exceed the available memory even for modest values of n.

If the matrices A_i, B_i, and C_i are sparse, then the time complexity of the outer step of Algorithm 4.1 is of order $\mathcal{O}(m\mathrm{MV})$, where MV stands for a matrix-vector multiplication by an $n \times n$ matrix.

8. Numerical examples. The following numerical results were obtained with Matlab 6.5.

In the first examples we use a two-parameter eigenvalue problem with known eigenpairs, which enables us to check the obtained results. The construction is similar to the one in [10], and therefore the details are omitted.

We take matrices

$$A_i = V_i F_i U_i, \qquad B_i = V_i G_i U_i, \qquad C_i = V_i H_i U_i \tag{8.1}$$

of dimension $n \times n$, where F_i, G_i, and H_i are complex diagonal matrices and U_i, V_i are random matrices for $i = 1, 2$. We select diagonal elements of matrices F_i, G_i, and H_i as complex numbers $\alpha + i\beta$, where α and β are uniformly distributed random numbers from the interval $(-0.5, 0.5)$. All the eigenvalues can be computed from the diagonal elements of F_i, G_i, and H_i for $i = 1, 2$.

Example 8.1. We compare different correction equations without preconditioning on matrices (8.1) of size $n = 100$. For the initial vectors we perturb the exact eigenvectors with a random perturbation of order 10^{-3}. In each step 2(b) of Algorithm 4.1 we take the Petrov triple with the smallest residual (4.3).

Table 8.1 contains the number of steps required for the residual (4.3) to become smaller than 10^{-8}. The other parameters are $l_{\max} = 10$, $l_{\min} = 2$, and $k_{\max} = 200$. We compared three two-sided correction equations without preconditioning:

- NP1—first-order correction equation (5.11), where $s_i \perp v_i$ and $t_i \perp u_i$;
- NP2—first-order correction equation (5.15), where $s_i \perp u_i$ and $t_i \perp v_i$. Although it is preconditioned, we treat this equation as an unpreconditioned one because the preconditioner is the identity.

TABLE 8.1

Comparison of three correction equations NP1, NP2, and NP3 without preconditioning for the initial vectors $\|u_i - x_{1i}\| = \mathcal{O}(10^{-3})$ and $\|v_i - y_{1i}\| = \mathcal{O}(10^{-3})$. GMRES: the number of steps used in GMRES for the approximate solution of the correction equation; Iterations: the number of outer iterations for convergence.

NP1		NP2		NP3	
GMRES	Iterations	GMRES	Iterations	GMRES	Iterations
90	> 200	90	> 200	180	50
95	46	95	36	190	25
99	3	99	3	199	5

TABLE 8.2

Comparison of three correction equations $P1$, $P2$, and $P3$ with preconditioning for initial vectors $\|u_i - x_{1i}\| = \mathcal{O}(10^{-3})e$ and $\|v_i - y_{1i}\| = \mathcal{O}(10^{-3})$. GMRES: the number of steps used in GMRES for the approximate solution of the correction equation; Iterations: the number of outer iterations for convergence.

P1		P2		P3	
GMRES	Iterations	GMRES	Iterations	GMRES	Iterations
1	63	1	63	1	99
2	70	2	59	2	36
4	28	4	28	4	24
8	6	8	6	8	6
15	4	15	4	15	3

- NP3—second-order correction equations (5.18) and (5.19).

The results in Table 8.1 indicate that the convergence is slow or we have no convergence at all if the correction equations are not solved accurately, and this happens as the number of GMRES steps gets closer to the size of the matrices. Let us remark that the number of GMRES steps for the second-order correction equation is larger because the size of the matrices is twice the size of the matrices in the first-order correction equations.

Example 8.2. For the second example we take the same initial vectors and parameters as in Example 8.1, but this time we use preconditioned correction equations. For a preconditioner we take (5.14). We compared the following three two-sided preconditioned correction equations:

- P1—preconditioned first-order correction equation NP1 from Example 8.1, where $s_i \perp v_i$ and $t_i \perp u_i$;
- P2—preconditioned first-order correction equation (5.13), where $s_i \perp u_i$, $t_i \perp v_i$, and where the second-order terms are small close to the eigenvalue;
- P3—preconditioned second-order correction equation NP3 from Example 8.1; see (5.20).

The results in Table 8.2 indicate that correction equations with preconditioners work better than the ones that are not preconditioned, and we have a fast convergence for a modest number of GMRES steps.

Example 8.3. In this example we use matrices (8.1) of size $n = 1000$. We take initial vectors $u_1 = u_2 = v_1 = v_2 = [1 \cdots 1]^T$ and parameters $l_{\max} = 15$ and $l_{\min} = 4$. Our goal is the eigenvalue closest to the origin. In step 2(b) of Algorithm 4.1 we pick the Petrov triple with the Petrov value closest to the target $(0, 0)$ until the residual ρ_k is less than $\varepsilon_{\text{change}} = 10^{-2.5}$. After that we take the Petrov triple with the smallest residual (4.3) until the residual is less than $5 \cdot 10^{-7}$.

Figure 8.1 shows the convergence plot for two-sided and one-sided correction equations P2 and P3 using various number of GMRES steps to solve the correction equation. One can see that once the residual becomes smaller than $\varepsilon_{\text{change}}$ (top horizontal dotted line in the figures) and we are close to the eigentriple, the number of GMRES steps determines the speed of the convergence.

There is no guarantee that the process will converge to the eigenvalue closest to the target. Table 8.3 shows the indices of the obtained eigenvalues if the eigenvalues are ordered by their distance from the target. This example shows that although the one-sided methods may converge faster than the two-sided methods (especially measured in number of matrix-vector multiplications), they often converge to an undesired eigenvalue.

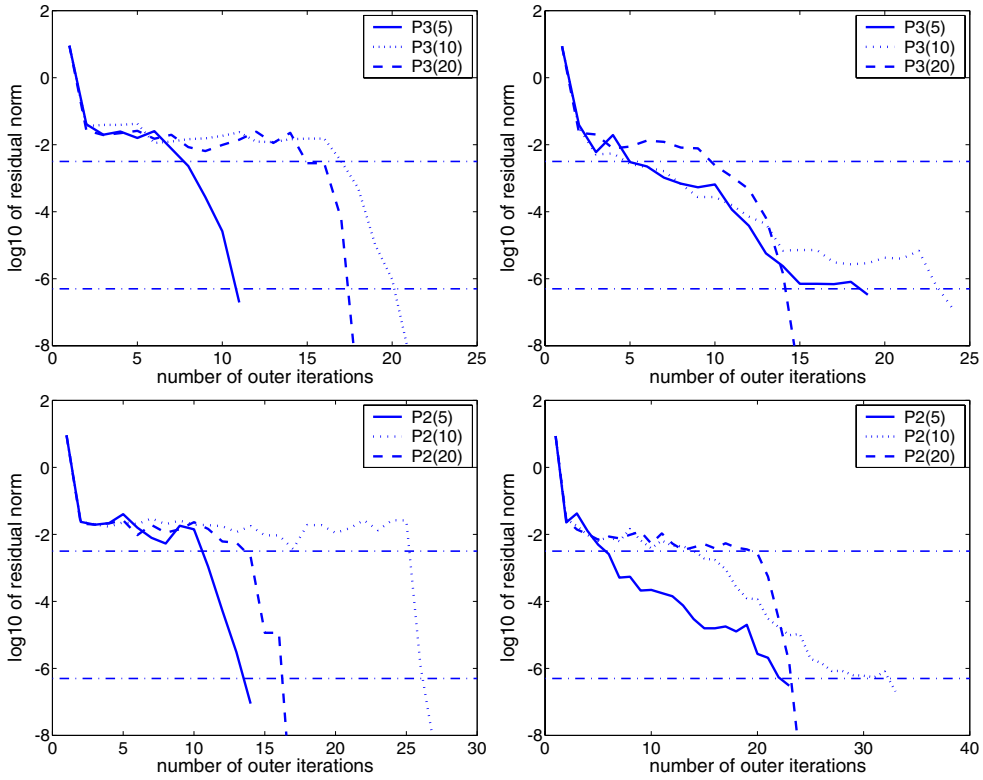


FIG. 8.1. *Convergence plot for the eigenvalue closest to* $(0,0)$ *for* $u_i = v_i = [1 \cdots 1]^T$. *The plots show the* $\log_{10}$ *of the residual norm* (4.3) *versus the outer iteration number for the Jacobi–Davidson type method using a correction equation with* 5 *(solid line),* 10 *(dotted line), and* 20 *(dashed line) GMRES steps to solve the correction equation. The correction equations are: two-sided* $P3$ *(top left), one-sided* $P3$ *(top right), two-sided* $P2$ *(bottom left), and one-sided* $P2$ *(bottom right).*

TABLE 8.3
Indices of the obtained eigenvalues from Figure 8.1.

GMRES	Two-sided P3	One-sided P3	Two-sided P2	One-sided P2
1	2	16	3	1
2	1	26	2	80
3	2	4	2	9

Example 8.4. In this example we test the selection technique from section 6 that enables us to compute more than one eigenvalue. The matrices are the same as in the previous example. Figure 8.2 shows a convergence plot for the first ten computed eigenvalues. For each eigenvalue we select the closest Petrov value to the origin until the residual becomes smaller than $\varepsilon_{\text{change}}$, and in the remaining steps we select the Petrov triple with the minimum residual. We consider only Petrov triples that satisfy the condition (6.1). The indices of the computed eigenvalues, ordered as they were obtained, are 1, 34, 4, 5, 2, 16, 3, 6, 9, and 12. The statistics in the following example show that the probability of a successful convergence is high if we carefully tune the parameters of the method.

Example 8.5. We use the same $n = 1000$ matrices as in Example 8.4. We test the

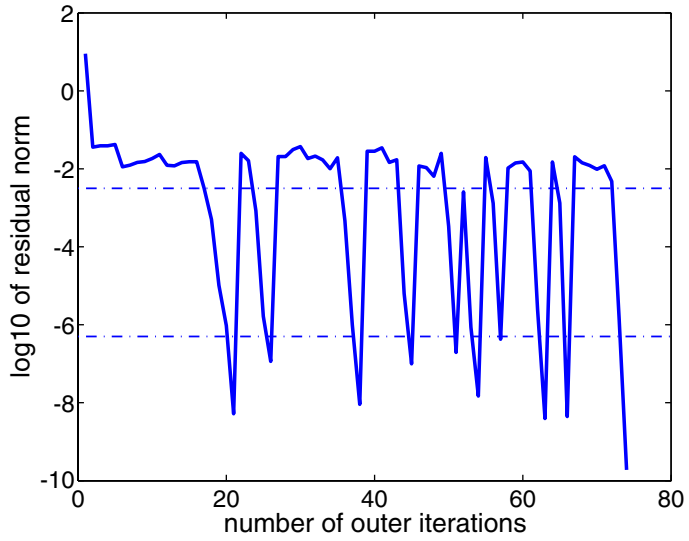


FIG. 8.2. *Convergence plot for the first ten computed eigenvalues using the selection technique from section* 6. *Used is correction equation* $P3$ *with* 15 *GMRES steps and parameters* $l_{\max} = 15$, $l_{\min} = 4$, *and* $\varepsilon_{\text{change}} = 10^{-2.5}$.

TABLE 8.4

Statistics of the Jacobi–Davidson type method using the same set of ten random initial vectors for computing the ten eigenvalues closest to the origin, using correction equation $P3$ *and a different number of GMRES steps and* $\varepsilon_{\text{change}}$. *The parameters are* $l_{\max} = 15$ *and* $l_{\min} = 4$; *maximum number of outer iterations is* 300. *GMRES: the number of steps used in GMRES for the approximate solution of the correction equation; In* 10 *(In* 50*): the average number of the computed eigenvalues among the* 10 *(*50*) closest eigenvalues to the origin; Conv.: the average number of computed eigenvalues; Iter.: the average number of outer iterations for convergence.*

Two-sided correction equation P3												
	$\varepsilon_{\text{change}} = 10^{-2}$				$\varepsilon_{\text{change}} = 10^{-3}$				$\varepsilon_{\text{change}} = 10^{-4}$			
GMRES	In 10	In 50	Conv.	Iter.	In 10	In 50	Conv.	Iter.	In 10	In 50	Conv.	Iter.
10	6.8	8.4	10.0	72.8	7.8	9.5	9.9	115.9	7.7	9.3	10.0	89.7
20	6.3	8.8	10.0	56.7	7.8	9.5	10.0	91.7	8.7	9.7	10.0	113.3
30	6.9	7.8	10.0	65.6	7.9	9.2	10.0	88.6	8.7	9.7	10.0	124.2
One-sided correction equation P3												
	$\varepsilon_{\text{change}} = 10^{-2}$				$\varepsilon_{\text{change}} = 10^{-3}$				$\varepsilon_{\text{change}} = 10^{-4}$			
GMRES	In 10	In 50	Conv.	Iter.	In 10	In 50	Conv.	Iter.	In 10	In 50	Conv.	Iter.
10	4.5	8.0	10.0	97.5	6.6	8.7	9.5	103.8	6.6	7.7	8.1	192.8
20	2.4	6.5	10.0	80.8	6.2	8.8	10.0	122.1	7.7	8.8	9.1	204.3
30	1.3	2.4	8.4	160.6	6.2	9.0	10.0	121.7	6.7	8.8	9.0	200.3

preconditioned correction equation P3 on the same set of ten random initial vectors. For each initial vector the goal was to compute the ten eigenvalues closest to the target, using the same approach as in the previous example. We set the maximum number of outer steps to 300 and use a different number of GMRES steps and a different $\varepsilon_{\text{change}}$.

The numbers in Table 8.4 show that the probability of computing the correct eigenvalues is high when the parameters are carefully chosen. If $\varepsilon_{\text{change}}$ is too small,

then in the first phase, when we select the closest Petrov value to the origin, the method requires too many iterations until the residual is smaller than $\varepsilon_{\text{change}}$. On the other hand, if $\varepsilon_{\text{change}}$ is too large, then the method is likely to converge fast, but to an unwanted eigenvalue. More GMRES steps may reduce the number of outer iterations and enlarge the probability, but we must keep in mind that the total amount of work is dependent on the number of matrix-vector multiplications, and thus roughly equal to the product of the number of GMRES steps and outer iterations. Also, if we use too many GMRES steps, then the correction equations are solved too accurately and the method requires more iterations until the residual is smaller than $\varepsilon_{\text{change}}$.

The results show that we can compute more eigenvalues close to the target if we use the two-sided method. The performance of the one-sided method is less optimal. The one-sided method usually requires more outer iterations, and situations where we have very slow convergence or no convergence at all occur more frequently.

Example 8.6. In the last example we study the three-point problem

$$y'' + (\lambda + \mu \cos x)y = 0 \tag{8.2}$$

with boundary conditions

$$y(0) = y(2.5) = y(5) = 0.$$

Instead of (8.2) we can study the two-parameter problem

$$y_i'' + (\lambda + \mu \cos x_i)y_i = 0, \quad i = 1, 2, \tag{8.3}$$

where $x_1 \in [0, 2.5]$, $x_2 \in [2.5, 5]$, and the boundary conditions are $y_1(0) = y_1(2.5) = 0$ and $y_2(2.5) = y_2(5) = 0$. One can see from the determinant

$$\begin{vmatrix} 1 & \cos(x_1) \\ 1 & \cos(x_2) \end{vmatrix} = \cos(x_2) - \cos(x_1)$$

that (8.3) is not right definite.

We can compute eigenvalues of (8.3) using finite differences. If we take $h = 1/(n-1)$, $x_{1i} = ih$, and $x_{2i} = x_{1i} + 2.5$ for $i = 1, \ldots, n$, then the $n \times n$ matrices that form the two-parameter problem are

$$\begin{gathered} A_1 = A_2 = \frac{1}{h^2}\,\text{tridiag}(1, -2, 1), \\ B_1 = B_2 = I, \\ C_1 = \text{diag}(\cos(x_{11}), \ldots, \cos(x_{1n})), \qquad C_2 = \text{diag}(\cos(x_{21}), \ldots, \cos(x_{2n})). \end{gathered} \tag{8.4}$$

The eigenfunctions for the six closest eigenvalues to $(0, 0)$ are shown in Figure 8.3.

Using finite differences and $n = 1000$, we test preconditioned correction equation P3 using the same set of 50 random initial vectors and various numbers of GMRES steps. The goal is to compute the ten closest eigenvalues to the target $(0, 0)$. Results in Table 8.5 show that it is possible to compute a selection of the closest eigenvalues to the target using the Jacobi–Davidson type method. It appears that the optimal solution in this case is to take a modest number of GMRES steps.

In this example the difference in the performance of the one-sided and the two-sided approaches is smaller than in Example 8.5. This happens because the matrices are real symmetric and therefore the left and right eigenvectors of real eigenvalues agree. The discretized problem (8.4) has complex eigenvalues as well, but the ones that we are interested in are all real.

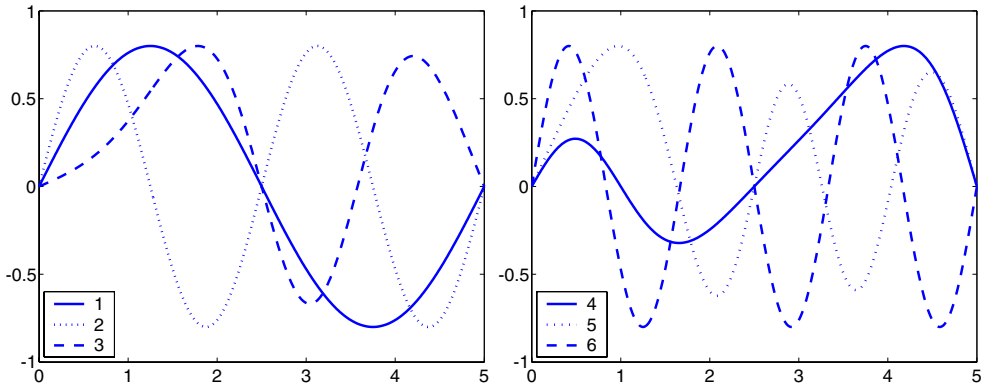


FIG. 8.3. *Eigenfunctions of the three-point boundary problem* (8.2) *for the six closest eigenvalues to* $(0,0)$*:* $(\lambda_1,\mu_1) = (-1.5790, 0)$*,* $(\lambda_2,\mu_2) = (-6.3145, 0)$*,* $(\lambda_3,\mu_3) = (-2.1197, 6.5418)$*,* $(\lambda_4,\mu_4) = (-5.1698, -5.4264)$*,* $(\lambda_5,\mu_5) = (-8.9898, 8.4441)$*, and* $(\lambda_6,\mu_6) = (-14.2019, 0)$*.*

TABLE 8.5

Statistics of the Jacobi–Davidson type method using the same set of ten random initial vectors for computing the ten closest eigenvalues to the origin using correction equation $P3$ *and different numbers of GMRES steps for problem* (8.4) *and* $n = 1000$*. The parameters are* $l_{\max} = 15$*,* $l_{\min} = 4$*, and* $\varepsilon_{\text{change}} = 10^{-2}$*. GMRES: the number of steps used in GMRES for the approximate solution of the correction equation; In* 10*: the average number of the computed eigenvalues among the ten closest eigenvalues to the origin; Iterations: the average number of outer iterations for convergence.*

Corr. equation	GMRES	In 10	Iterations
Two-sided P3	5	10.0	86.2
Two-sided P3	10	10.0	48.9
Two-sided P3	20	9.9	42.2
Two-sided P3	30	10.0	50.8
One-sided P3	5	10.0	70.5
One-sided P3	10	9.8	50.7
One-sided P3	20	10.0	68.2
One-sided P3	30	9.9	90.3

9. Conclusions. We have presented a new Jacobi–Davidson type method for the nonsingular two-parameter eigenvalue problem. This problem is a very challenging one, where we have to use many available techniques to be successful: a two-sided subspace approach, preconditioning, selection techniques instead of deflating, and the use of a target.

Numerical examples show that the two-sided subspace approach is often more expensive, but also more reliable. An additional advantage of the two-sided approach is that during the process we have approximate left and right eigenvectors, and hence in principle (see [11] for details) an approximation to the condition number of the eigenvalue to which we are converging.

The new method can compute selected eigenpairs without good initial approximations, and it can tackle very large two-parameter problems, especially if the matrices A_i, B_i, and C_i are sparse. In such situations, preconditioning is of great importance.

Let us also mention that Algorithms 2.3 and 4.1 both offer a simple generalization to multiparameter problems with more than two parameters.

Acknowledgments. The authors are grateful to the referees for careful reading of the paper and several helpful comments.

REFERENCES

[1] F. V. Atkinson, *Multiparameter spectral theory*, Bull. Amer. Math. Soc., 74 (1968), pp. 1–27.

[2] F. V. Atkinson, *Multiparameter Eigenvalue Problems*, Academic Press, New York, 1972.

[3] 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*, SIAM, Philadelphia, 2000.

[4] Z. Bohte, *Numerical solution of some two-parameter eigenvalue problems*, in Anton Kuhelj Memorial Volume, Slovenian Academie of Science and Art, Ljubljana, Slovenia, 1982, pp. 17–28.

[5] A. Bunse-Gerstner, R. Byers, and V. Mehrmann, *Numerical methods for simultaneous diagonalization*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 927–949.

[6] N. Cottin, *Dynamic model updating—A multiparameter eigenvalue problem*, Mech. Systems Signal Process., 15 (2001), pp. 649–665.

[7] L. Fox, L. Hayes, and D. F. Mayers, *The double eigenvalue problems*, in Topics in Numerical Analysis, Proceedings of the Royal Irish Academy Conference on Numerical Analysis, J. J. H. Miller, ed., Academic Press, New York, 1972, pp. 93–112.

[8] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[9] M. E. Hochstenbach and G. L. G. Sleijpen, *Two-sided and alternating Jacobi–Davidson*, Linear Algebra Appl., 358 (2003), pp. 145–172.

[10] M. E. Hochstenbach and B. Plestenjak, *A Jacobi–Davidson type method for a right definite two-parameter eigenvalue problem*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 392–410.

[11] M. E. Hochstenbach and B. Plestenjak, *Backward error, condition numbers, and pseudospectrum for the multiparameter eigenvalue problem*, Linear Algebra Appl., 375 (2003), pp. 63–81.

[12] X. Ji, *Numerical solution of joint eigenpairs of a family of commutative matrices*, Appl. Math. Lett., 4 (1991), pp. 57–60.

[13] T. Košir, *Finite dimensional multiparameter spectral theory: The nonderogatory case*, Linear Algebra Appl., 212/213 (1994), pp. 45–70.

[14] J. R. Kuttler and V. G. Sigillito, *Eigenvalues of the Laplacian in two dimensions*, SIAM Rev., 26 (1984), pp. 163–193.

[15] B. C. Lesieutre, A. V. Mamishev, Y. Du, E. Keskiner, M. Zahn, and G. C. Verghese, *Forward and inverse parameter estimation algorithms of interdigital dielectrometry sensors*, IEEE Trans. Dielectrecs Elect. Insul., 8 (2001), pp. 577–588.

[16] M. R. Osborne, *Iterative procedures for solving finite-difference approximations to separable partial differential equation*, Comput. J., 6 (1963), pp. 93–99.

[17] B. Plestenjak, *A continuation method for a right definite two-parameter eigenvalue problem*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1163–1184.

[18] B. Plestenjak, *A continuation method for a weakly elliptic two-parameter eigenvalue problem*, IMA J. Numer. Anal., 21 (2001), pp. 199–216.

[19] 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.

[20] T. Slivnik and G. Tomšič, *A numerical method for the solution of two-parameter eigenvalue problems*, J. Comput. Appl. Math., 15 (1986), pp. 109–115.

[21] G. W. Stewart and J. G. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.

[22] P. Van Dooren, *A generalized eigenvalue approach for solving Riccati equations*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 121–135.

[23] H. Volkmer, *Multiparameter Problems and Expansion Theorems*, Lecture Notes in Math. 1356, Springer-Verlag, New York, 1988.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 2, pp. 498–521

A SUBSPACE APPROXIMATION METHOD FOR THE QUADRATIC EIGENVALUE PROBLEM*

U. B. HOLZ†, G. H. GOLUB‡, AND K. H. LAW§

Abstract. Quadratic eigenvalue problems involving large matrices arise frequently in areas such as the vibration analysis of structures, micro-electro-mechanical systems (MEMS) simulation, and the solution of quadratically constrained least squares problems. The typical approach is to solve the quadratic eigenvalue problem using a mathematically equivalent linearized formulation, resulting in a doubled dimension and, in many cases, a lack of backward stability.

This paper introduces an approach to solving the quadratic eigenvalue problem directly without linearizing it. Perturbation subspaces for block eigenvector matrices are used to reduce the modified problem to a sequence of problems of smaller dimension. These perturbation subspaces are shown to be contained in certain generalized Krylov subspaces of the n-dimensional space, where n is the undoubled dimension of the matrices in the quadratic problem. The method converges at least as fast as the corresponding Taylor series, and the convergence can be accelerated further by applying a block generalization of the quadratically convergent Rayleigh quotient iteration. Numerical examples are presented to illustrate the applicability of the method.

Key words. quadratic eigenvalue problems, generalized Krylov subspaces, subspace approximation method, block Rayleigh quotient iteration

AMS subject classifications. 15A18, 65F15

DOI. 10.1137/S0895479803423378

1. Introduction.

The quadratic eigenvalue problem

$$(\lambda^2 \mathrm{M} + \lambda \mathrm{C} + \mathrm{K})\mathbf{x} = 0 \tag{1.1}$$

commonly arises during the solution of systems of second order ordinary differential equations found in scientific and engineering applications. Gohberg, Lancaster, and Rodman [7] and Lancaster [14] provided an extensive theoretical background on quadratic and other polynomial eigenvalue problems. For a current review of numerical methods for quadratic eigenvalue problems along with a broad discussion of application areas, see Tisseur and Meerbergen [19]. The most common approach is to expand (1.1), for example, as

$$\begin{pmatrix} 0 & \mathrm{N} \\ -\mathrm{K} & -\mathrm{C} \end{pmatrix} \mathbf{z} = \lambda \begin{pmatrix} \mathrm{N} & 0 \\ 0 & \mathrm{M} \end{pmatrix} \mathbf{z} \quad \text{or} \quad \begin{pmatrix} -\mathrm{C} & -\mathrm{K} \\ \mathrm{N} & 0 \end{pmatrix} \mathbf{w} = \lambda \begin{pmatrix} \mathrm{M} & 0 \\ 0 & \mathrm{N} \end{pmatrix} \mathbf{w}, \tag{1.2}$$

where N is any nonsingular matrix. Not only does the linearized problem have twice the dimension of the quadratic problem, but also, in general, even if a backward stable method is used for the linear eigenvalue problem, that stability is not guaranteed for the quadratic eigenvalue problem, as shown by Tisseur [18]. This paper introduces a method that tackles the quadratic eigenvalue problem directly using subspace approximation and perturbation techniques.

*Received by the editors February 13, 2003; accepted for publication (in revised form) April 5, 2004; published electronically January 12, 2005.
http://www.siam.org/journals/simax/26-2/42337.html
†Mathematics Department, Stanford University, Stanford, CA 94305 (ubholz@earthlink.net).
‡Scientific Computing and Computational Mathematics Program, Stanford University, Stanford, CA 94305 (golub@sccm.stanford.edu).
§Department of Civil and Environmental Engineering, Stanford University, Stanford, CA 94305 (law@stanford.edu).

1.1. Subspace approximation method for linear eigenvalue problems. Zhang, Golub, and Law [21] presented a generalized Krylov subspace method for the perturbed symmetric standard eigenvalue problem, $(\mathrm{A}+\Delta\mathrm{A})\mathbf{x} = \lambda\mathbf{x}$, given the known solution for $\mathrm{A}\mathbf{x} = \lambda\mathbf{x}$. The method is based on the following theorem.

THEOREM 1.1 (see [21]). *Assume $Q = [Q_1\ Q_2]$ is orthogonal, with each Q_i representing an eigenspace, and assume $\Lambda_1 = Q_1^T \mathrm{A} Q_1 = \lambda_1 \mathrm{I}$. Let $\Lambda_2 = Q_2^T \mathrm{A} Q_2$, $E = Q_2(\Lambda_2 - \lambda_1 \mathrm{I})Q_2^T$, and $F = E\Delta\mathrm{A}$. Let Q_1^m be the eigenspace of $\mathrm{A} + \Delta\mathrm{A}$ (as a perturbation of Q_1) obtained by the mth order Taylor series expansion. Then Q_1^m belongs to the subspace $\mathcal{K}(E, F, Q_1, m)$, where*

$$\mathcal{K}(E, F, Q_1, m) = \mathcal{R}([P_0(E, F)Q_1, \ldots, P_m(E, F)Q_1]).$$

Here $P_k(E, F)$ is the space spanned by all the homogeneous polynomials in E and F of order k, and $\mathcal{R}(Y)$ denotes the range of Y.

The method computes the spaces $\mathcal{K}(E, F, Q_1, m)$, $m = 1, 2, 3, \ldots$, and solves the reduced problems in these spaces until convergence of the eigenpairs. The method is at least as fast as the convergence of the corresponding Taylor polynomials. In the current paper this subspace approximation concept is employed for solving the quadratic eigenvalue problem.

1.2. The perturbed quadratic eigenvalue problem. Consider computing a few eigenpairs for the perturbed problem

$$(\lambda^2(\mathrm{M} + \Delta\mathrm{M}) + \lambda(\mathrm{C} + \Delta\mathrm{C}) + (\mathrm{K} + \Delta\mathrm{K}))\mathbf{x} = 0, \tag{1.3}$$

assuming that corresponding eigenpairs for the unperturbed problem, $(\lambda^2\mathrm{M} + \lambda\mathrm{C} + \mathrm{K})\mathbf{x} = 0$, are known [11]. For the special case of the quadratic eigenvalue problem discussed in this paper, we consider the case with $\Delta\mathrm{M} = \mathrm{C} = \Delta\mathrm{K} = 0$ and at least one of M and K nonsingular. That is, we regard the matrix C in (1.1) as a perturbation, and we consider the quadratic eigenvalue problem as a special case of the perturbed quadratic eigenvalue problem. The analysis holds for those perturbations which result in a convergent Taylor series for the eigenvector matrix, and may be extended using homotopy [11] to future work involving arbitrarily large perturbations.

This paper is organized as follows. In section 2 a block perturbation form of (1.3) is introduced and a subspace approximation theorem is proved. Then in section 3 the computation of perturbation subspaces is described, both in terms of generalized Krylov subspaces and in terms of smaller, directly computed subspaces. Section 4 gives a first order error analysis and develops a stopping criterion. In section 5 a hybrid algorithm is developed using perturbation subspaces and block Rayleigh quotients, and in section 6 the complexity of the subspace approximation computations is considered. Section 7 relates the subspace approximation method to existing methods. Finally, section 8 illustrates the subspace approximation method, using numerical examples drawn from structural dynamics applications.

The numerical examples are performed using MATLAB 6.1.0 on a 1 gigahertz Sun Blade 2000 with 2 gigabytes of main memory, running Solaris 8.

2. Block quadratic equation. Given M, C, and K in $\mathbb{R}^{\mathrm{n}\times\mathrm{n}}$, with M nonsingular, let $P(\lambda, t) = \lambda^2\mathrm{M} + \lambda t\mathrm{C} + \mathrm{K}$ for $\lambda \in \mathbb{C}$ and $0 \le t \le 1$. Consider the eigenvalue problem

$$P(\lambda(t), t)\mathbf{x}(t) = 0, \quad t \in [0, 1]. \tag{2.1}$$

Because M is nonsingular for t in $[0,1]$, there exist continuous eigenvalue paths $\lambda_1(t), \lambda_2(t), \ldots, \lambda_{2n}(t)$. (See, e.g., Ahlfors [1, Section 8.2].) If, instead, M is singular but K is nonsingular, all the theory of this section still applies to the problem rearranged as $P(\mu(t), t)\mathbf{x}(t) = 0$, $t \in [0,1]$, with $P(\mu, t) \equiv \mathrm{M} + \mu t\mathrm{C} + \mu^2\mathrm{K}$ and $\lambda(t) = \frac{1}{\mu(t)}$ for $\mu(t) \neq 0$.

When the eigenvalues $\lambda(t)$ of interest are nondefective, it is useful to compute a subspace that contains approximations to the associated eigenspaces. We write the block version of (2.1) as

$$\mathrm{M}X(t)\Lambda^2(t) + t\mathrm{C}X(t)\Lambda(t) + \mathrm{K}X(t) = 0, \tag{2.2}$$

where we know solutions at $t = 0$ and seek solutions at $t = 1$. The idea of the subspace approximation method is to compute subspaces that contain the ranges of the Taylor approximations to $X(t)$ and then to solve the reduced quadratic eigenvalue problems in these subspaces to obtain the approximate eigenpairs for (2.1) on the whole space.

The following notational conventions are used in our discussion:

1. The superscript (j) denotes the jth derivative with respect to t, at $t = 0$ unless t is otherwise specified. For example, if $Q(t)$ is a matrix function of t, then $Q^{(j)}$ is its jth derivative at $t = 0$, and $Q^{(j)}(t_0)$ is its jth derivative at $t = t_0$.
2. $\|\cdot\|$ denotes the Euclidean norm unless otherwise stated.
3. $X_j(t)$ is the jth Taylor approximation about $t = 0$ to the function $X(t)$ in (2.2).

2.1. Convergence of the block Taylor series. When discussing Taylor approximations it is, of course, important to explore issues of convergence. In this section we examine convergence of the block Taylor series for the eigenvector matrix $X(t)$ in the case of nondefective eigenvalues (including simple eigenvalues), and we extend the discussion to the case of defective eigenvalues and mention ideas for the case of distinct but clustered eigenvalues.

Suppose we know a nondefective eigenvalue λ of multiplicity p for (2.1) at $t = 0$, along with a corresponding $n \times p$ right eigenvector matrix X_0. Writing the associated eigenvalue paths as $\lambda_1(t), \ldots, \lambda_p(t)$, in (2.2), $X(t)$ is an $n \times p$ matrix function of t with $X(0) = X_0$, and $\Lambda(t)$ is a $p \times p$ matrix function of t whose eigenvalues are $\lambda_1(t), \ldots, \lambda_p(t)$. No assumptions are made here regarding the normalization of $X(t)$ since the results in this section are independent of normalization.

For nondefective λ the matrix function $X(t)$ can be taken to have a convergent Taylor series as follows. Consider a standard linearized form for (2.1), such as

$$A(t)\mathbf{z}(t) \equiv \begin{pmatrix} -t\mathrm{M}^{-1}\mathrm{C} & -\mathrm{M}^{-1}\mathrm{K} \\ \mathrm{I} & 0 \end{pmatrix} \begin{pmatrix} \lambda(t)\mathbf{x}(t) \\ \mathbf{x}(t) \end{pmatrix} = \lambda(t) \begin{pmatrix} \lambda(t)\mathbf{x}(t) \\ \mathbf{x}(t) \end{pmatrix}. \tag{2.3}$$

When λ_0 is a nondefective eigenvalue of some multiplicity p for (2.3) at $t = t_0$, the corresponding eigenspace projection $P(t)$, also called the total projection for the λ-group eigenvalues of $A(t)$, is holomorphic (i.e., possesses a derivative everywhere) in a neighborhood of t_0 in $\mathbb{C}$ (see Kato [12, Section II.1.4]). It is shown in [12, Section II.4.2] that if a projection $P(t)$ is holomorphic in some domain D containing t_0, then there is a transformation function $U(t)$ satisfying the following: (1) $U(t)^{-1}$ exists and both $U(t)$ and $U(t)^{-1}$ are holomorphic on D, (2) $U(t)P(t_0)U(t)^{-1} = P(t)$ on D, and (3) $U(t_0) = I$. It follows that if the p columns of Z_0 form a basis for $P(t_0)$, then the p columns of $Z(t) = U(t)Z_0$ form a holomorphic basis for $P(t)$. Now taking $Z_0 = \binom{\lambda_0 X_0}{X_0}$ and writing $Z(t) = \binom{Z_1(t)}{Z_2(t)}$, the block form of (2.3) is $A(t)Z(t) = Z(t)\Lambda(t)$,

and with some manipulation, yields $\mathrm{M}Z_2(t)\Lambda^2(t) + t\mathrm{C}Z_2(t)\Lambda(t) + \mathrm{K}Z_2(t) = 0$, where $Z_2(t_0) = X_0$. Since $Z_2(t)$ is holomorphic and of full rank, taking

$$X(t) = Z_2(t)W(t) \tag{2.4}$$

for any nonsingular holomorphic $p \times p$ matrix $W(t)$ satisfying $W(t_0) = \mathrm{I}_p$ gives a holomorphic block eigenvector matrix $X(t)$. In particular if $W(t)$ is holomorphic on the whole complex plane and ρ is the convergence radius about t_0 of the Taylor series for $Z(t)$, then the convergence radius of the Taylor series for $X(t)$ about t_0 is at least ρ. (A lower bound for ρ may be computed using majorization series, described in [12, Section II.3.1]; however, this is very expensive, involving explicit formation of a $2n$ matrix inverse, a $2n$ pseudoinverse, and several $2n$ matrix norms.)

Although multiple nondefective eigenvalues are, in fact, uncommon, much of the above discussion remains relevant in the case of a cluster of close eigenvalues with λ_0 as their arithmetic mean. Also, even when λ_0 is a defective eigenvalue of $A(t)$ at t_0, the total projection $P(t)$ onto the associated invariant subspace is holomorphic. Thus if $P_{\mathrm{tot}}(t)$ is the total projection for the sum of the invariant subspaces associated with p eigenvalue paths $\lambda_{i_1}(t), \lambda_{i_2}(t), \ldots, \lambda_{i_p}(t)$, then P_{tot} can be analytically continued as t goes from 0 to 1, as long as no other eigenvalue paths intersect these. $Z(t)$ can, therefore, be analytically continued, as can $X(t)$ if it is defined by (2.4), although, in this case, $X(t)$ may be rank-deficient.

2.2. A subspace approximation theorem. To specify the perturbation subspaces we first require the definition of a generalized Krylov subspace.

DEFINITION 2.1. *For $B_1, B_2, \ldots, B_k \in \mathbb{C}^{N\times N}$, and $X \in \mathbb{C}^{N\times p}$, $0 < p \le N$, let $\mathcal{S}_j(B_1, B_2, \ldots, B_k, X)$, abbreviated $\mathcal{S}_j(X)$ when the B_i's are understood, denote the jth generalized Krylov subspace generated by $B_1, B_2, \ldots, B_k$ applied j times to X, i.e.,*

$$\mathcal{S}_j(X) = \sum_{p\le j} \mathrm{range}(B_{i_1} B_{i_2} \cdots B_{i_p} X).$$

As an equivalent definition, let $\mathcal{S}_0(B_1, B_2, \ldots, B_k, X) = \mathrm{range}(X)$, and, for $j > 0$, if the columns of X_{j-1} form a basis for $\mathcal{S}_{j-1}(B_1, B_2, \ldots, B_k, X)$, let

$$\mathcal{S}_j(B_1, B_2, \ldots, B_k, X) = \mathrm{range}([B_1X_{j-1} \;\; B_2X_{j-1} \;\; \cdots \;\; B_kX_{j-1} \;\; X_{j-1}]).$$

The following result gives generalized Krylov subspaces containing the ranges of the Taylor approximations to $X(t)$. These spaces are specified explicitly in terms of the coefficient matrices and an operator F which depends on a complement to the range of X and which need only be applied rather than kept available in matrix form. In the next section this theorem will be used to obtain a sequence of subspaces from which eigenvector and eigenvalue approximations can be computed.

THEOREM 2.2. *Let $V \in \mathbb{C}^{n\times(n-p)}$ be such that* $\mathrm{range}(V) + \mathrm{range}(X_0) = \mathbb{C}^n$, *and let F be an $n \times n$ matrix satisfying*

$$FP(\lambda_0, 0)V = V. \tag{2.5}$$

Then for all t, for all $j \ge 0$, $\mathrm{range}(X_j(t)) \subseteq \mathcal{S}_j(F\mathrm{M}, F\mathrm{C}, X_0)$.

Proof. Since $\mathcal{S}_j(X_0) \subseteq \mathcal{S}_{j+1}(X_0)$ for all $j \ge 0$, it is sufficient to show that

$$\mathrm{range}(X^{(j)}) \subseteq \mathcal{S}_j(X_0) \quad \forall j \ge 0. \tag{2.6}$$

By the definition of $\mathcal{S}_j(X_0)$, (2.6) is true for $j = 0$. We proceed by induction on j. Assume (2.6) holds for all $j < k$, where $k > 0$. Then $\text{range}(X^{(i)}) \subseteq \mathcal{S}_j(X_0)$ for all i and j such that $0 \le i \le j < k$. Taking the kth derivative with respect to t of (2.2), setting $t = 0$, and applying F yields

$$FP(\lambda_0,0)X^{(k)} = -\sum_{r=0}^{k-1}\binom{k}{r}\mathrm{M}X^{(r)}(\Lambda^2)^{(k-r)} - \sum_{r=0}^{k-1}k\binom{k-1}{r}\mathrm{C}X^{(r)}\Lambda^{(k-1-r)}. \tag{2.7}$$

If the columns of Q_{k-1} form a basis for $\mathcal{S}_{k-1}(X_0)$, then $\text{range}(X^{(j)}) \subseteq \text{range}(Q_{k-1})$ for all $j < k$, so the range of the right-hand side of (2.7) is contained in

$$\text{range}([F\mathrm{M}Q_{k-1}, F\mathrm{C}Q_{k-1}]),$$

which is in turn contained in $\mathcal{S}_k(X_0)$. Let the columns of Q_k be a basis for $\mathcal{S}_k(X_0)$ of size $n \times p_k$. Then $FP(\lambda_0,0)X^{(k)} = Q_kT_k$ for some $p_k \times p$ matrix T_k. Writing $X^{(k)} = Q^{(k)}_{(1)} + Q^{(k)}_{(2)}$, where the columns of $Q^{(k)}_{(1)}$ are in $\text{range}(X_0)$ and the columns of $Q^{(k)}_{(2)}$ are in $\text{range}(V)$, and using (2.5), we have $Q^{(k)}_{(2)} = Q_kT_k$. Hence

$$\text{range}(X^{(k)}) \subseteq \text{range}\left([Q^{(k)}_{(1)}Q^{(k)}_{(2)}]\right) \subseteq \text{range}([X_0Q_k]) = \mathcal{S}_k(X_0).$$

Thus (2.6) holds for all $j \ge 0$, which proves the theorem. ∎

3. Subspace computations.

3.1. Applying the subspace approximation theorem. If $V = [\mathbf{v}_1 \ \cdots \ \mathbf{v}_{n-p}]$ satisfies $\text{range}(V) + \text{range}(X_0) = \mathbb{C}^n$, then clearly V is of full rank and $\text{range}(V) \cap \text{range}(X_0) = 0$. Also, the $n - p$ columns of $P(\lambda_0,0)V$ are linearly independent and, therefore, form a basis for $\text{range}(P(\lambda_0,0))$. Now let $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_p$ be a basis for $\text{range}(P(\lambda_0,0))^\perp$. For any p vectors $\mathbf{w}_j \in \mathbb{C}^n$ there is an associated matrix F satisfying (2.5), specified by

$$\begin{aligned} F(P(\lambda_0,0)\mathbf{v}_i) &= \mathbf{v}_i, \quad i = 1,2,\ldots,n-p, \\ F\mathbf{y}_j &= \mathbf{w}_j, \quad j = 1,2,\ldots,p. \end{aligned} \tag{3.1}$$

The condition

$$V \in \mathbb{C}^{n\times(n-p)}, \quad \text{range}(V) + \text{range}(X_0) = \mathbb{C}^n \tag{3.2}$$

thus implies the existence of a matrix F satisfying (2.5) that is uniquely determined by $\mathbf{w}_1, \ldots, \mathbf{w}_p$. Also, (3.2) alone uniquely determines $FP(\lambda_0,0)$ since $FP(\lambda_0,0)V = V$ and $FP(\lambda_0,0)X_0 = 0$.

To apply Theorem 2.2 we must first specify V in some way. A natural choice is to let V satisfy $\text{range}(V) = \text{range}(X_0)^\perp$. Let P_0^+ be the pseudoinverse of $P(\lambda_0,0)$. Then $P_0^+P(\lambda_0,0)$ is the orthonormal projection into $\text{null}(P)^\perp = \text{range}(V)$, and if $P(\lambda_0,0)^*\mathbf{y} = 0$, then $P_0^+\mathbf{y} = 0$. Hence the F satisfying (2.5) determined by this V and $\mathbf{w}_j = 0$, $j = 1,2,\ldots,p$, is exactly P_0^+. (Note that in the version of the subspace theorem for the standard symmetric eigenvalue problem this choice of F gives Theorem 1.1 (see [21]).) Now we can compute the subspaces $\mathcal{S}_j$ by solving appropriate least squares problems. The problems $\min\|P(\lambda_0,0)\mathbf{v} - \mathbf{y}\|$ are rank-deficient, but because $\text{null}(P(\lambda_0,0)) = \text{range}(X_0)$, deflation using Householder transformations can be used to obtain equivalent full rank problems. Alternatively, the least squares

problems may be considered only nearly rank-deficient numerically, in which case we can choose to solve them directly by decomposing $P(\lambda_0, 0)$. This is an ill-conditioned problem, resulting in large error components in range(X_0) that must then be removed to gain acceptable solutions.

In the following sections, two algorithms for subspace computation are considered: one to be used when the left eigenvector matrix of the unperturbed problem is unknown, and the other to be used when the left eigenvector matrix is known. In the first situation the subspaces may grow exponentially, but in the second the subspaces grow linearly in the number of desired eigenvalues.

3.2. Computing the full perturbation subspaces. Suppose we have $X_1, X_2, \ldots, X_{j-1}$ such that

$$\mathcal{S}_{j-1} = \text{range}(X_0) \oplus \text{range}(X_1) \oplus \cdots \oplus \text{range}(X_{j-1}).$$

Then, writing $Y_{j-1} = [X_0\ X_1\ \cdots\ X_{j-1}]$, $\mathcal{S}_j = \text{range}([F\text{M}Y_{j-1}, F\text{C}Y_{j-1}, Y_{j-1}])$. Since, for $k < j-1$, the ranges of $F\text{M}X_k$ and $F\text{C}X_k$ are contained in range(Y_{k+1}), which is contained in range(Y_{j-1}), it follows that

$$\mathcal{S}_j = \text{range}([F\text{M}X_{j-1}, F\text{C}X_{j-1}, Y_{j-1}]).$$

Setting $\mathcal{S}_{-1} = \emptyset$ and $\mathcal{S}_0 = \text{range}(X_0)$, we proceed as follows to compute $\mathcal{S}_j$ for $j > 0$.

ALGORITHM 3.1. *This algorithm computes* $\widehat{W}$ *such that* $\mathcal{S}_j = \mathcal{S}_{j-1} + \text{range}(\widehat{W})$.

0. *Let* X_{j-1} *satisfy* $\mathcal{S}_{j-1} = \mathcal{S}_{j-2} \oplus \text{range}(X_{j-1})$. *That is,* X_{j-1} *is full rank, such that the span of its columns added to* $\mathcal{S}_{j-2}$ *gives the space* $\mathcal{S}_{j-1}$ *with* $\dim(\mathcal{S}_{j-1}) = \dim(\mathcal{S}_{j-2}) + \text{rank}(X_{j-1})$.

1. *Let the columns of* W *form a basis for* range$([\text{M}X_{j-1}, \text{C}X_{j-1}])$.

2. *Solve the least squares problem* $\min \|P(\lambda_0, 0)\hat{\mathbf{w}}_i - \mathbf{w}_i\|$ *for each column of* W *to get* $\widehat{W}$. *(If solving directly, first project* $\mathbf{w}_i$ *into* $\mathcal{S}_0^{\perp}$ *so that the part of the solution not in the range of* X_0 *will be numerically significant.)*

3. $\mathcal{S}_j = \mathcal{S}_{j-1} + \text{range}(\widehat{W})$ *because* $\text{range}([\widehat{W}\ X_0]) = \text{range}([FW\ X_0])$.

To add range$(\widehat{W})$ to $\mathcal{S}_{j-1}$, modified Gram–Schmidt is used to get the orthonormal basis $[X_0\ X_1\ \cdots\ X_j]$ for range$([X_0\ X_1\ \cdots\ X_{j-1}\ \widehat{W}])$, so that

$$\mathcal{S}_j = \text{range}([X_0\ X_1\ \cdots\ X_j]) = \mathcal{S}_{j-1} \oplus \text{range}(X_j).$$

Note that this algorithm computes the generalized Krylov subspace by powers. It will be interesting, in future work, to consider computing the space using other polynomials. Also note that at each step the dimension of the space may triple (as opposed to the doubling that may occur when size $2n$ linearized forms such as (1.2) are used). The following section discusses a way to avoid exponential subspace growth if possible.

3.3. Directly computing derivative subspaces. The equations leading to the proof of Theorem 2.2 suggest a way to compute the derivatives $X^{(k)}$ directly within the generalized Krylov subspace. As mentioned above, the results in section 2.2 are independent of the normalization of $X(t)$. Now assume the normalization condition

$$X_0{}^* X(t) = \text{I}, \quad t \in [0, 1]. \tag{3.3}$$

In addition, assume we know a matrix of left eigenvectors $W_0 \in \mathbb{C}^{n \times p}$ associated with λ_0 at time $t = 0$, i.e., $W_0{}^* P(\lambda_0, 0) = 0$, such that

$$2\lambda_0 W_0{}^* \text{M} X_0 \text{ is nonsingular.} \tag{3.4}$$

In some instances a value of W_0 is clear from the properties of the problem. For example, when M and K are symmetric, if C is skew-symmetric, then $W_0 = X_0$, and if C is symmetric, then $W_0 = \overline{X}_0$, the conjugate of X_0. Condition (3.4) is guaranteed, as the following lemma shows.

LEMMA 3.2. *Let λ_0 be an eigenvalue of geometric multiplicity p for $P(\lambda, 0)$, and let X_0 and W_0 be associated full rank right and left eigenvector matrices. Then* (3.4) *holds if and only if λ_0 is nondefective.*

Proof. Let $P(\lambda, 0) = E(\lambda)\Gamma(\lambda)F(\lambda)$ be the Smith canonical decomposition of P (see Wilkinson [20, pp. 19–20]) so that $E(\lambda)$ and $F(\lambda)$ are nonsingular $n \times n$ matrices with determinants independent of λ, and $\Gamma(\lambda) = \operatorname{diag}(a_j(\lambda))$, where the functions $a_j(\lambda)$ are monic polynomials in λ such that each polynomial is a factor of the next one, i.e., $a_1(\lambda) \mid a_2(\lambda) \mid \cdots \mid a_n(\lambda)$. Since λ_0 is of geometric multiplicity p,

$$\begin{aligned} a_j(\lambda_0) &\neq 0 \quad \text{for } j \leq n-p, \\ a_j(\lambda_0) &= 0 \quad \text{for } j > n-p. \end{aligned} \tag{3.5}$$

Write $f(\lambda) = W_0^* P(\lambda, 0) X_0 = (W_0 E(\lambda))\Gamma(\lambda)(F(\lambda)X_0)$, and let $\widehat{W}_0(\lambda) = E(\lambda)^* W_0$ and $\widehat{X}_0(\lambda) = F(\lambda)X_0$. $\widehat{W}_0(\lambda)$ and $\widehat{X}_0(\lambda)$ are of full rank for all values of λ, and from (3.5) and the facts

$$\widehat{W}_0(\lambda_0)^* \Gamma(\lambda_0) = 0, \qquad \Gamma(\lambda_0)\widehat{X}_0(\lambda_0) = 0,$$

it follows that $\widehat{W}_0(\lambda_0)^* = [0 \ w_1(\lambda_0)]$ and $\widehat{X}_0(\lambda_0) = [0 \ x_1(\lambda_0)]^T$, where $w_1(\lambda_0)$ and $x_1(\lambda_0)$ are nonsingular $p \times p$ matrices. Then

$$\begin{aligned} W_0{}^*(2\lambda_0 \mathrm{M} + \mathrm{C})X_0 = f'(\lambda_0) &= \widehat{W}_0(\lambda_0)^* \Gamma'(\lambda_0)\widehat{X}_0(\lambda_0) \\ &= w_1(\lambda_0) \operatorname{diag}(a'_{n-p+1}(\lambda_0), \ldots, a_n{}'(\lambda_0)) x_1(\lambda_0). \end{aligned} \tag{3.6}$$

Condition (3.4) holds exactly when $a_j{}'(\lambda_0) \neq 0$ for all $j > n-p$, which is true if and only if λ_0 has algebraic multiplicity p. ∎

To get $X^{(k)}$ directly we again differentiate (2.2),

$$P(\lambda_0, 0)X^{(k)} + \sum_{r=0}^{k-1} \binom{k}{r} \mathrm{M} X^{(r)} (\Lambda^2)^{(k-r)} + \sum_{r=0}^{k-1} k \binom{k-1}{r} \mathrm{C} X^{(r)} \Lambda^{(k-1-r)} = 0, \tag{3.7}$$

and, extracting the terms in $\Lambda^{(k)}$ and using the fact that $k\binom{k-1}{j} = (k-j)\binom{k}{j}$,

$$P(\lambda_0, 0)X^{(k)} + (2\lambda_0 \mathrm{M})X_0\Lambda^{(k)} = -\mathrm{M}X_0 \sum_{l=1}^{k-1} \binom{k}{l} \Lambda^{(l)}\Lambda^{(k-l)} - \mathrm{M} \sum_{j=1}^{k-1} \binom{k}{j} X^{(k-j)} (\Lambda^2)^{(j)}. \tag{3.8}$$

Let V_k denote the right-hand side of (3.8). Then, premultiplying (3.8) by $W_0{}^*$,

$$(2\lambda_0 W_0^* \mathrm{M} X_0)\Lambda^{(k)} = W_0^* V_k. \tag{3.9}$$

If all the values of $\Lambda^{(j)}$ and $X^{(j)}$ are known for $j < k$, we can compute V_k in a straightforward manner using its definition, so (3.9) may be solved uniquely for $\Lambda^{(k)}$. The columns of $-2\lambda_0 \mathrm{M} X_0 \Lambda^{(k)} + V_k$ are in the range of $P(\lambda_0, 0)$. Let Z_k be any solution to

$$P(\lambda_0, 0)Z_k = -2\lambda_0 \mathrm{M} X_0 \Lambda^{(k)} + V_k. \tag{3.10}$$

Then for some $v_k \in \mathbb{C}^{p\times p}$, $X^{(k)} = Z_k + X_0 v_k$. By (3.3) $X_0^* Z_k + X_0^* X_0 v_k = X_0^* X^{(k)} = 0$, so $v_k = -X_0^* Z_k$ and

$$X^{(k)} = Z_k - X_0(X_0^* Z_k), \tag{3.11}$$

i.e., $X^{(k)} = (\mathrm{I} - X_0 X_0^*) Z_k$, the projection of the columns of Z_k into $\mathrm{range}(X_0)^\perp$. Thus we compute $X^{(k)}$ as follows.

ALGORITHM 3.3. *Given* $\Lambda^{(0)} = \lambda_0 \mathrm{I}$, $X^{(0)} = X_0$, W_0, *and* $\Lambda^{(1)}, \Lambda^{(2)}, \ldots, \Lambda^{(k-1)}$, $X^{(1)}, X^{(2)}, \ldots, X^{(k-1)}$, *this algorithm computes* $\Lambda^{(k)}$ *and* $X^{(k)}$.

$X_0 = X^{(0)}$; $X_1 = X^{(k-1)}$;
$Z_{\mathrm{M}} = X_0 \sum_{l=1}^{k-1} \binom{k}{l} \Lambda^{(l)} \Lambda^{(k-l)}$; $Z_{\mathrm{C}} = k\lambda_0 X_1$;
for $j = 1\colon k-1$ /* *compute sums on the right-hand side of* (3.8) */
  $X_2 = X_1$; $X_1 = X^{(k-1-j)}$;
  $L = \Lambda^{(j)}$; $L_1 = \sum_{l=0}^{j} \binom{j}{l} \Lambda^{(l)} \Lambda^{(j-l)}$; $c = \binom{k}{j}$;
  $Z_{\mathrm{M}} = Z_{\mathrm{M}} + c X_2 L_1$;
  $Z_{\mathrm{C}} = Z_{\mathrm{C}} + (k-j) c X_1 L$;
end
$V_k = -(\mathrm{M} Z_{\mathrm{M}} + \mathrm{C} Z_{\mathrm{C}})$;
$Z = 2\lambda_0 \mathrm{M} X_0$; $Z = Z - X_0(X_0^* Z)$;
Solve $W_0^* Z \Lambda^{(k)} = W_0^* V_k$ *for* $\Lambda^{(k)}$.
Solve $P(\lambda_0, 0) X = -Z\Lambda^{(k)} + V_k$ *for* X.
$X^{(k)} = X - X_0(X_0^* X)$.

To compute the subspace associated with s distinct nonconjugate eigenvalues we perform the above procedure for each eigenvalue independently and then combine the s computed subspaces to get the desired space. For a conjugate pair of eigenvalues it is enough to compute the subspace for one of the two since the bases determining the two subspaces are conjugate.

3.4. Real arithmetic in subspace computations. Under certain conditions the computation of the subspaces can be arranged in a way that involves only real arithmetic, as can be seen from the following lemma, the proof of which is a straightforward case-by-case check, left to the reader.

LEMMA 3.4. *Let* M *and* K *be symmetric, and let the quadratic eigenvalue problem* $(\lambda^2 \mathrm{M} + \mathrm{K})\mathbf{x} = 0$ *have only real eigenvectors* $\mathbf{x}$ *associated with an imaginary nondefective eigenvalue* $\lambda_0 = i\omega_0 \neq 0$. *Let* $\Lambda^{(k)}$ *and* $X^{(k)}$ *be as in section* 2.2, *and suppose* $X_0^T \mathrm{M} X_0$ *is nonsingular. Assume* C *is a nonzero matrix. Then*

$$\begin{cases} \Lambda^{(k)} \in i\mathbb{R}^{p\times p} \text{ and } X^{(k)} \in \mathbb{R}^{n\times p}, & \text{when } k \text{ is even,} \\ \Lambda^{(k)} \in \mathbb{R}^{p\times p} \text{ and } X^{(k)} \in i\mathbb{R}^{n\times p}, & \text{when } k \text{ is odd.} \end{cases}$$

Instead of looking at $\Lambda^{(k)}$ and $X^{(k)}$, let us look at the imaginary parts when the matrices are imaginary and the real parts when the matrices are real. Write

$$\Lambda^{(k)} = i^{(1-k \bmod 2)} \Omega_k, \qquad X^{(k)} = i^{k \bmod 2} Y_k, \tag{3.12}$$

where, by the lemma, Ω_k and Y_k are real matrices. Since the perturbation subspaces are determined by the sets $\mathrm{range}(X^{(k)}) = \mathrm{range}(Y_k)$ it suffices to work with (Ω_k, Y_k)

rather than $(\Lambda_k, X^{(k)})$. Substituting (3.12) into (3.8) gives

$$\begin{aligned} & i^{k \bmod 2} P(\lambda_0, 0) Y_k + (2i\omega_0 \mathrm{M}) Y_0 i^{(1-k \bmod 2)} \Omega_k \\ & \quad = -\mathrm{M} Y_0 \sum_{l=1}^{k-1} \binom{k}{l} i^{(1-l \bmod 2)+(1-(k-l) \bmod 2)} \Omega_l \Omega_{k-l} \\ & \quad\quad -\mathrm{M} \sum_{j=1}^{k-1} \left(\binom{k}{j} i^{(k-j) \bmod 2} Y_{k-j} \sum_{l=0}^{j} \binom{j}{l} i^{(1-l \bmod 2)+(1-(j-l) \bmod 2)} \Omega_l \Omega_{j-l} \right) \\ & \quad\quad -\mathrm{C} \sum_{j=0}^{k-1} k \binom{k-1}{j} i^{(k-1-j) \bmod 2} Y_{k-1-j} i^{1-j \bmod 2} \Omega_j, \end{aligned} \tag{3.13}$$

which can be rewritten as

$$\begin{aligned} & P(\lambda_0, 0) Y_k + (-1)^{k-1} 2\omega_0 \mathrm{M} Y_0 \Omega_k \\ & \quad = -\mathrm{M} Y_0 \sum_{l=1}^{k-1} \binom{k}{l} (-1)^{(k-1)(l-1)} \Omega_l \Omega_{k-l} \\ & \quad\quad -\mathrm{M} \sum_{j=1}^{k-1} \left(\binom{k}{j} Y_{k-j} \sum_{l=0}^{j} \binom{j}{l} (-1)^{(kj-1)(l(j-1)-1)} \Omega_l \Omega_{j-l} \right) \\ & \quad\quad -\mathrm{C} \sum_{j=0}^{k-1} k \binom{k-1}{j} (-1)^{(k-1)(j-1)} Y_{k-1-j} \Omega_j. \end{aligned} \tag{3.14}$$

Just as in section 3.3, we can now compute Ω_k and Y_k directly, this time using only real arithmetic.

3.5. Solving the reduced problem. If $Q \in \mathbb{C}^{n \times r}$ gives an orthonormal basis for the subspace $\mathcal{S}$, let

$$M_{\mathrm{proj}} = Q^* \mathrm{M} Q, \quad C_{\mathrm{proj}} = Q^* \mathrm{C} Q, \quad \text{and} \quad K_{\mathrm{proj}} = Q^* \mathrm{K} Q, \tag{3.15}$$

and consider the solutions $(\lambda_i, \mathbf{y}_i)$, $i = 1, 2, \ldots, 2r$, to

$$(\lambda^2 M_{\mathrm{proj}} + \lambda C_{\mathrm{proj}} + K_{\mathrm{proj}})\mathbf{y} = 0. \tag{3.16}$$

The approximate solutions $(\lambda_i, \mathbf{x}_i) = (\lambda_i, Q\mathbf{y}_i)$ are exactly the eigenpairs for the quadratic problem with the operators M, C, and K replaced by their projections onto $\mathcal{S}$. See Hochstenbach and van der Vorst [9] for alternative ways of getting approximate solutions from a given subspace.

The reduced quadratic problem (3.16) has complex matrices M_{proj}, C_{proj}, and K_{proj}, resulting in a complex linearized problem. These matrices can, instead, be forced to be real using the fact that, for $\mathbf{w} \in \mathbb{C}^r$, $Q\mathbf{w} = [\mathrm{real}(Q)\ \ \mathrm{imag}(Q)]\binom{\mathbf{w}}{i\mathbf{w}}$, which implies $\mathrm{range}(Q) \subseteq \mathrm{range}([\mathrm{real}(Q)\ \ \mathrm{imag}(Q)])$. If Q_1 is a matrix whose columns form an orthonormal basis for $[\mathrm{real}(Q)\ \ \mathrm{imag}(Q)]$, then $\mathcal{S} \subseteq \mathrm{range}(Q_1)$. Thus using Q_1 instead of Q in (3.15) results in a reduced problem involving only real matrices, and the best eigenspace approximations in $\mathrm{range}(Q_1)$ are at least as good as those in $\mathcal{S}$. The corresponding linear problem is of a dimension up to twice that of the linear problem formed using Q, and the question is whether it is cheaper to find the basis Q_1, project M, C, and K onto $\mathrm{range}(Q_1)$, solve the resulting real linearized problem, and form the approximate eigenpairs, rather than working with the complex basis Q.

TABLE 3.1
Flop comparison between real and complex bases.

	Flops, using Q	Flops, using [real(Q) imag(Q)]
Forming basis	done	$4nr$
Projecting	$6(2nr)(r+n)$	$4nr(2r+n)$
Solving linearized problem	$6(25)(r^3)$	$25(2r)^3$
Computing approximate eigenvectors	$6(2r)(2ns)$	$4r(2ns)$
Total	$12nr^2+12n^2r+150r^3+24nrs$	$4nr+8nr^2+4n^2r+200r^3+8nrs$

A simplified operation count provides an answer. Assume a real scalar operation counts as one flop and a complex one counts as six. (This is the convention used in MATLAB 5, for example.) The comparison for computing s nonconjugate eigenpairs is given in Table 3.1, showing that it is better to use Q_1 when

$$12nr^2+12n^2r+150r^3+24nrs > 4nr+8nr^2+4n^2r+200r^3+8nrs,$$

which holds exactly when the cardinality r of the complex basis satisfies

$$0 < r < \frac{n}{25}+\frac{1}{25}\sqrt{101n^2+200ns-50n}. \tag{3.17}$$

Usually we are interested in the s eigenvalues of smallest (or largest) magnitude, and the first idea might be to choose as our approximations the s smallest (or largest) λ_i and corresponding $\mathbf{x}_i$ for the projected problem. However, unless the eigenvalues are known to satisfy a minimax or interlacing property, for example, in the case of overdamped systems (see, e.g., Duffin [6]) or conservative gyroscopic systems (see [11, Chapter 4] and Bauchau [2]), a further check is needed to eliminate spurious values. In the next section an eigenvalue error estimate is introduced that will be used to weed out these poor approximations.

4. First order error and stopping criterion.

4.1. Error in eigenvalues. For any given matrices M, C, and K, with M nonsingular, suppose the pair $(\mu,\mathbf{y})$ is an approximation to an eigenpair $(\lambda_i,\mathbf{x}_i)$ for

$$(\lambda^2\mathrm{M}+\lambda\mathrm{C}+\mathrm{K})\mathbf{x}=0, \tag{4.1}$$

with λ_i a simple eigenvalue and μ not equal to any eigenvalue of (4.1), and suppose we know the associated residual $\mathbf{r}=(\mu^2\mathrm{M}+\mu\mathrm{C}+\mathrm{K})\mathbf{y}$. Consider the problem

$$\left(\lambda(\epsilon)^2\mathrm{M}+\lambda(\epsilon)\mathrm{C}+\mathrm{K}-\left(1-\epsilon\frac{\|\mathbf{y}\|}{\|\mathbf{r}\|}\right)\frac{\mathbf{r}\mathbf{u}^T}{\mathbf{u}^T\mathbf{y}}\right)\mathbf{x}(\epsilon)=0, \tag{4.2}$$

where $\mathbf{u}$ is any vector such that $\mathbf{u}^T\mathbf{y}\neq 0$. It is straightforward to check that $(\mu,\mathbf{y})$ is a solution to (4.2) at $\epsilon=0$. Since μ is not an eigenvalue of (4.1), $\mathbf{z}=\mathbf{y}$ is the unique solution to $(\mu^2\mathrm{M}+\mu\mathrm{C}+\mathrm{K})\mathbf{z}=\mathbf{r}$, and any nonzero vector $\hat{\mathbf{y}}$ satisfying

$$(\mu^2\mathrm{M}+\mu\mathrm{C}+\mathrm{K})\hat{\mathbf{y}}-\mathbf{r}\left(\frac{\mathbf{u}^T\hat{\mathbf{y}}}{\mathbf{u}^T\mathbf{y}}\right)=0$$

must be a multiple of $\mathbf{y}$. Thus μ is a simple eigenvalue, so for all sufficiently small ϵ the solution $(\lambda(\epsilon), \mathbf{x}(\epsilon))$ with $\mathbf{x}(\epsilon)^* \mathbf{x}_0 = 1$ exists, and we can write the Taylor series

$$\begin{aligned} \lambda(\epsilon) &= \mu + \epsilon\dot{\lambda}(0) + \epsilon^2 \frac{\ddot{\lambda}(0)}{2} + \cdots, \\ \mathbf{x}(\epsilon) &= \frac{\mathbf{y}}{\|\mathbf{y}\|} + \epsilon\dot{\mathbf{x}}(0) + \epsilon^2 \frac{\ddot{\mathbf{x}}(0)}{2} + \cdots. \end{aligned} \tag{4.3}$$

Substituting (4.3) into (4.2), and using the fact that the coefficient of the first power of ϵ (and, in fact, that of each power of ϵ) on the left-hand side of (4.2) is zero,

$$\left(2\mu\dot{\lambda}(0)\mathrm{M} + \dot{\lambda}(0)\mathrm{C} + \frac{\|\mathbf{y}\|}{\|\mathbf{r}\|}\frac{\mathbf{r}\mathbf{u}^T}{\mathbf{u}^T\mathbf{y}}\right)\frac{\mathbf{y}}{\|\mathbf{y}\|} + \left(\mu^2\mathrm{M} + \mu\mathrm{C} + \mathrm{K} - \frac{\mathbf{r}\mathbf{u}^T}{\mathbf{u}^T\mathbf{y}}\right)\dot{\mathbf{x}}(0) = 0,$$

and

$$\dot{\lambda}(0)(2\mu\mathrm{M} + \mathrm{C})\frac{\mathbf{y}}{\|\mathbf{y}\|} + \frac{\mathbf{r}}{\|\mathbf{r}\|} + (\mu^2\mathrm{M} + \mu\mathrm{C} + \mathrm{K})\dot{\mathbf{x}}(0) - \mathbf{r}\left(\frac{\mathbf{u}^T\dot{\mathbf{x}}(0)}{\mathbf{u}^T\mathbf{y}}\right) = 0. \tag{4.4}$$

Setting

$$\mathbf{u} = (\mu^2\mathrm{M} + \mu\mathrm{C} + \mathrm{K})^T\mathbf{w} \tag{4.5}$$

for any $\mathbf{w}$ satisfying

$$\mathbf{w}^T\mathbf{r} \neq 0 \tag{4.6}$$

and premultiplying (4.4) by $\mathbf{w}^T$ gives $\dot{\lambda}(0)\mathbf{w}^T(2\mu\mathrm{M} + \mathrm{C})\frac{\mathbf{y}}{\|\mathbf{y}\|} + \frac{\mathbf{w}^T\mathbf{r}}{\|\mathbf{r}\|} = 0$; thus

$$\dot{\lambda}(0) = -\frac{\|\mathbf{y}\|}{\|\mathbf{r}\|}\left(\frac{\mathbf{w}^T\mathbf{r}}{\mathbf{w}^T(2\mu\mathrm{M} + \mathrm{C})\mathbf{y}}\right)$$

and

$$\lambda(\epsilon) - \mu = -\epsilon - \frac{\|\mathbf{y}\|}{\|\mathbf{r}\|}\left(\frac{\mathbf{w}^T\mathbf{r}}{\mathbf{w}^T(2\mu\mathrm{M} + \mathrm{C})\mathbf{y}}\right) + O(\epsilon^2) \quad \text{as } \epsilon \to 0. \tag{4.7}$$

Next observe that at $\epsilon = \frac{\|\mathbf{r}\|}{\|\mathbf{y}\|}$, λ_i is by assumption a simple eigenvalue of (4.2), so if $\frac{\|\mathbf{r}\|}{\|\mathbf{y}\|}$ is small enough we have

$$|\lambda_i - \mu| = \frac{|\mathbf{w}^T\mathbf{r}|}{|\mathbf{w}^T(2\mu\mathrm{M} + \mathrm{C})\mathbf{y}|} + O\left(\left(\frac{\|\mathbf{r}\|}{\|\mathbf{y}\|}\right)^2\right) \quad \text{as } \frac{\|\mathbf{r}\|}{\|\mathbf{y}\|} \to 0. \tag{4.8}$$

Then a reasonable criterion for a solution pair $(\mu, \mathbf{y})$ to be acceptable is

$$\max\left(\frac{|\mathbf{w}^T\mathbf{r}|}{|\mathbf{w}^T(2\mu\mathrm{M} + \mathrm{C})\mathbf{y}|}, \left(\frac{\|\mathbf{r}\|}{\|\mathbf{y}\|}\right)^2\right) < \mu\,\mathrm{tol} \tag{4.9}$$

for some tolerance tol, i.e., the relative error in the eigenvalue is on the order of tol. A good choice of $\mathbf{w}$ is clearly $\mathbf{w} = \bar{\mathbf{r}}$ since this $\mathbf{w}$ fails to satisfy (4.6) only when $\mathbf{r} = 0$, in which case the approximate solution is, of course, acceptable. Then (4.9) becomes

$$\frac{\|\mathbf{r}\|^2}{\min(|\mathbf{r}^*(2\mu\mathrm{M} + \mathrm{C})\mathbf{y}|, \|\mathbf{y}\|^2)} < \mu\,\mathrm{tol}. \tag{4.10}$$

Now suppose we want approximations of the form $\mathbf{y}_i = W\mathbf{z}_i$, with $W \in \mathbb{C}^{n\times r}$ of full rank and $\mathbf{z}_i$ satisfying $(\mu_i^2 \mathrm{M}_{\mathrm{proj}} + \mu_i \mathrm{C}_{\mathrm{proj}} + \mathrm{K}_{\mathrm{proj}})\mathbf{z}_i = 0$, where $\mathrm{M}_{\mathrm{proj}} = W^*\mathrm{M}W$, $\mathrm{C}_{\mathrm{proj}} = W^*\mathrm{C}W$, and $\mathrm{K}_{\mathrm{proj}} = W^*\mathrm{K}W$. In other words suppose we are interested in pairs $(\mu_i, \mathbf{z}_i)$ resulting from solving a reduced quadratic eigenvalue problem as in section 3.5. Using (3.17), the following is an algorithm to solve the reduced quadratic eigenvalue problem and to select approximate solutions.

ALGORITHM 4.1. *Given a full rank matrix $W \in \mathbb{C}^{n\times r}$, this algorithm computes approximations to the $s < 2r$ eigenvalues of* (4.1) *smallest in magnitude, along with corresponding eigenvector approximations.*

1. *If r satisfies condition* (3.17)*, compute a real orthonormal basis W_1 for* range([real(W) imag(W)])*, and set $W = W_1$.*

2. *Form $\mathrm{M}_{\mathrm{proj}} = W^*\mathrm{M}W$, $\mathrm{C}_{\mathrm{proj}} = W^*\mathrm{C}W$, and $\mathrm{K}_{\mathrm{proj}} = W^*\mathrm{K}W$.*

3. *Linearize and use methods for small, dense matrices to compute the eigenvalues $\mu_1, \mu_2, \ldots, \mu_{2r}$ and corresponding eigenvectors $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{2r}$ of*

$$(\mu^2 \mathrm{M}_{\mathrm{proj}} + \mu \mathrm{C}_{\mathrm{proj}} + \mathrm{K}_{\mathrm{proj}})\mathbf{z} = 0.$$

4. *For $i = 1, 2, \ldots, 2r$, compute $\mathbf{y}_i = W\mathbf{z}_i$.*

5. *Use criterion* (4.10) *to select the approximate eigenpairs $(\mu_{i_j}, \mathbf{y}_{i_j})$, $j = 1, 2, \ldots, s$, with the smallest values of* $\mathrm{relerrs}_i \equiv \frac{1}{|\mu_i|}\frac{\|\mathbf{r}_i\|^2}{\min(|\mathbf{r}_i^*(2\mu\mathrm{M}+\mathrm{C})\mathbf{y}_i|, \|\mathbf{y}\|^2)}$.

In steps 4 and 5 of Algorithm 4.1 the computation is done only once for each complex conjugate pair. Note that if W is orthogonal, $\|\mathbf{y}_i\| = \|\mathbf{z}_i\|$. Also note that in step 3 we are linearizing the reduced problem in order to solve it, resulting in a possible lack of backward stability, as discussed in section 1. Alternative approaches, such as applying the subspace approximation method recursively in order to minimize this difficulty, have been suggested and remain to be explored.

4.2. Error in eigenvectors. If instead of (4.5) we set $\mathbf{u}$ to be the conjugate

$$\mathbf{u} = \overline{\mathbf{y}} \tag{4.11}$$

and define $\mathbf{v} = (\mu^2\mathrm{M} + \mu\mathrm{C} + \mathrm{K})^{-1}(2\mu\mathrm{M} + \mathrm{C})\mathbf{y}$, we can show the following.

LEMMA 4.2. *If $\angle(\mathbf{y}, \mathbf{v}) \neq 0$, then $\angle(\mathbf{y}, \mathbf{x}_i) = \angle(\mathbf{y}, \mathbf{v}) + O((\frac{\|\mathbf{r}\|}{\|\mathbf{y}\|})^2)$ as $\frac{\|\mathbf{r}\|}{\|\mathbf{y}\|} \to 0$.*

For the proof of the lemma two other results are needed.

PROPOSITION 4.3. $\|\dot{\mathbf{x}}(0)\|^2 = (\frac{\|\mathbf{y}\|}{\|\mathbf{r}\|})^2 \frac{1}{\left|\left(\frac{\mathbf{y}}{\|\mathbf{y}\|}\right)^*\mathbf{v}\right|^2}\ (\|\mathbf{v}\|^2 - |(\frac{\mathbf{y}}{\|\mathbf{y}\|})^*\mathbf{v}|^2)$.

Proof. Applying $(\mu^2\mathrm{M} + \mu\mathrm{C} + \mathrm{K})^{-1}$ to (4.4),

$$\dot{\mathbf{x}}(0) = -\dot{\lambda}(0)\frac{\mathbf{v}}{\|\mathbf{y}\|} + \left(\frac{1}{\|\mathbf{r}\|} - \frac{\mathbf{u}^T\dot{\mathbf{x}}(0)}{\mathbf{u}^T\mathbf{y}}\right)\mathbf{y},$$

and since $\mathbf{y}^*\dot{\mathbf{x}}(0) = 0$, it follows that $0 = -\dot{\lambda}(0)\frac{\mathbf{y}^*\mathbf{v}}{\|\mathbf{y}\|} + (\frac{1}{\|\mathbf{r}\|} - \frac{\mathbf{u}^T\dot{\mathbf{x}}(0)}{\mathbf{u}^T\mathbf{y}})\|\mathbf{y}\|^2$, so

$$\dot{\mathbf{x}}(0) = -\dot{\lambda}(0)\frac{\mathbf{v}}{\|\mathbf{y}\|} + \frac{\dot{\lambda}(0)\mathbf{y}^*\mathbf{v}}{\|\mathbf{y}\|^3}\mathbf{y}.$$

Now let $\mathbf{z}$ be a nonzero vector satisfying

$$\mathbf{z}^*\left(\mu^2\mathrm{M} + \mu\mathrm{C} + \mathrm{K} - \frac{\mathbf{r}\mathbf{u}^T}{\mathbf{u}^T\mathbf{y}}\right) = 0.$$

Then $\mathbf{z}^*(\mu^2\mathrm{M}+\mu\mathrm{C}+\mathrm{K}) = (\frac{\mathbf{z}^*\mathbf{r}}{\mathbf{u}^T\mathbf{y}})\mathbf{u}^T$, so $\mathbf{z}^* = \alpha\mathbf{u}^T(\mu^2\mathrm{M}+\mu\mathrm{C}+\mathrm{K})^{-1}$ for some α. Applying $\mathbf{z}^*$ to (4.4) yields

$$\dot\lambda(0)\mathbf{z}^*(2\mu\mathrm{M}+\mathrm{C})\frac{\mathbf{y}}{\|\mathbf{y}\|}+\frac{\mathbf{z}^*\mathbf{r}}{\|\mathbf{r}\|}=0,$$

$$\dot\lambda(0) = -\frac{\|\mathbf{y}\|}{\|\mathbf{r}\|}\frac{\mathbf{z}^*\mathbf{r}}{\mathbf{z}^*(2\mu\mathrm{M}+\mathrm{C})\mathbf{y}} = -\frac{\|\mathbf{y}\|}{\|\mathbf{r}\|}\frac{\mathbf{u}^T\mathbf{y}}{\mathbf{u}^T\mathbf{v}}.$$

Hence

$$\begin{aligned}\dot{\mathbf{x}}(0) &= -\frac{\|\mathbf{y}\|}{\|\mathbf{r}\|}\frac{\mathbf{u}^T\mathbf{y}}{\mathbf{u}^T\mathbf{v}}\left(-\frac{\mathbf{v}}{\|\mathbf{y}\|}+\frac{\mathbf{y}^*\mathbf{v}}{\|\mathbf{y}\|^3}\mathbf{y}\right)\\ &= -\frac{1}{\|\mathbf{r}\|}\frac{\mathbf{u}^T\mathbf{y}}{\mathbf{u}^T\mathbf{v}}\left(-\mathbf{v}+\left[\left(\frac{\mathbf{y}}{\|\mathbf{y}\|}\right)^*\mathbf{v}\right]\frac{\mathbf{y}}{\|\mathbf{y}\|}\right).\end{aligned}$$

Using (4.11),

$$\begin{aligned}\|\dot{\mathbf{x}}(0)\|^2 &= \left|\frac{1}{\|\mathbf{r}\|}\frac{\mathbf{u}^T\mathbf{y}}{\mathbf{u}^T\mathbf{v}}\right|^2\left\|-\mathbf{v}+\left[\left(\frac{\mathbf{y}}{\|\mathbf{y}\|}\right)^*\mathbf{v}\right]\frac{\mathbf{y}}{\|\mathbf{y}\|}\right\|^2\\ &= \left(\frac{\|\mathbf{y}\|}{\|\mathbf{r}\|}\right)^2\left|\frac{\|\mathbf{y}\|}{\mathbf{y}^*\mathbf{v}}\right|^2\left(\|\mathbf{v}\|^2-\frac{(\mathbf{v}^*\mathbf{y})(\mathbf{y}^*\mathbf{v})}{\|\mathbf{y}\|^2}-\frac{(\mathbf{y}^*\mathbf{v})(\mathbf{v}^*\mathbf{y})}{\|\mathbf{y}\|^2}+\frac{|\mathbf{y}^*\mathbf{v}|^2}{\|\mathbf{y}\|^2}\right)\\ &= \left(\frac{\|\mathbf{y}\|}{\|\mathbf{r}\|}\right)^2\frac{1}{|(\frac{\mathbf{y}}{\|\mathbf{y}\|})^*\mathbf{v}|^2}\left(\|\mathbf{v}\|^2-\left|\left(\frac{\mathbf{y}}{\|\mathbf{y}\|}\right)^*\mathbf{v}\right|^2\right). \qquad \square\end{aligned}$$

PROPOSITION 4.4. *If* $\dot{\mathbf{x}}(0)\neq 0$,

$$\angle(\mathbf{x}(\epsilon),\mathbf{y}) = \cos^{-1}\frac{1}{\sqrt{1+\epsilon^2\|\dot{\mathbf{x}}(0)\|^2}}+O(\epsilon^2) \quad \textit{as } \epsilon\to 0.$$

Proof. The proof is elementary calculus using the Taylor expansion of $f(w) = \cos^{-1}(w^{-1/2})$ in the appropriate interval.

Proof of Lemma 4.2. $\angle(\mathbf{y},\mathbf{v})\neq 0$ exactly when $|(\frac{\mathbf{y}}{\|\mathbf{y}\|})^*\mathbf{v}|\neq\|\mathbf{v}\|$, so, from Proposition 4.3,

$$\|\dot{\mathbf{x}}(0)\|^2 = \left(\frac{\|\mathbf{y}\|}{\|\mathbf{r}\|}\right)^2\frac{1}{\left|\left(\frac{\mathbf{y}}{\|\mathbf{y}\|}\right)^*\mathbf{v}\right|^2}\left(\|\mathbf{v}\|^2-\left|\left(\frac{\mathbf{y}}{\|\mathbf{y}\|}\right)^*\mathbf{v}\right|^2\right)\neq 0,$$

and applying Proposition 4.4 at $\epsilon = \frac{\|\mathbf{r}\|}{\|\mathbf{y}\|}$ we have

$$\begin{aligned}\angle(\mathbf{x}_i,\mathbf{y}) &= \cos^{-1}\frac{1}{\sqrt{1+\frac{1}{\left|\left(\frac{\mathbf{y}}{\|\mathbf{y}\|}\right)^*\mathbf{v}\right|^2}(\|\mathbf{v}\|^2-|(\frac{\mathbf{y}}{\|\mathbf{y}\|})^*\mathbf{v}|^2)}}+O\left(\left(\frac{\|\mathbf{r}\|}{\|\mathbf{y}\|}\right)^2\right) \quad \text{as } \frac{\|\mathbf{r}\|}{\|\mathbf{y}\|}\to 0\\ &= \cos^{-1}\frac{\left|\left(\frac{\mathbf{y}}{\|\mathbf{y}\|}\right)^*\mathbf{v}\right|}{\|\mathbf{v}\|}+O\left(\left(\frac{\|\mathbf{r}\|}{\|\mathbf{y}\|}\right)^2\right) \quad \text{as } \frac{\|\mathbf{r}\|}{\|\mathbf{y}\|}\to 0\\ &= \angle(\mathbf{y},\mathbf{v})+O\left(\left(\frac{\|\mathbf{r}\|}{\|\mathbf{y}\|}\right)^2\right) \quad \text{as } \frac{\|\mathbf{r}\|}{\|\mathbf{y}\|}\to 0. \qquad \square\end{aligned}$$

Computing $\mathbf{v}$ to examine this eigenvector error estimate at each step would be expensive; instead it is useful to calculate the estimate at the end of the computation to look at the final quality of the computed eigenvectors.

5. A hybrid method. Because the perturbation subspaces are constructed to contain the ranges of the Taylor series for the eigenspaces, the subspace approximations discussed above yield, within their convergence radius, eigenpair approximations that converge at least as well as the corresponding Taylor series, in other words at least linearly. To accelerate this convergence we would like to switch, at an appropriate stage, to a generalization of the quadratically convergent Rayleigh quotient iteration.

5.1. Block Rayleigh quotient iteration. Suppose that the vectors $X = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_s]$ form a basis for a space spanned by approximate right eigenvectors of the problem (4.1), and define a block Rayleigh quotient of (4.1) for X to be any $s \times s$ matrix Λ satisfying

$$X^*\mathrm{M}X\Lambda^2 + X^*\mathrm{C}X\Lambda + X^*\mathrm{K}X = 0. \tag{5.1}$$

One possible block generalization of Lancaster's Rayleigh quotient iteration (RQI) [13] can be described as follows.

ALGORITHM 5.1 (Block RQI 1). *This algorithm performs general block Rayleigh quotient iterations, starting with any X_1, to compute approximate eigenpairs for* (4.1). *For $l = 1, 2, 3, \ldots$*

0. *Given* range(X_l), *an approximate span of right eigenvectors for* (4.1).
1. *Find Λ_l (not unique) such that $(X_l^*\mathrm{M}X_l)\Lambda_l^2 + (X_l^*\mathrm{C}X_l)\Lambda_l + (X_l^*\mathrm{K}X_l) = 0$.*
2. *Solve $\mathrm{M}Y_{l+1}\Lambda_l^2 + \mathrm{C}Y_{l+1}\Lambda_l + \mathrm{K}Y_{l+1} = X_l$ for Y_{l+1}.*
3. *Let W be a basis for* range(Y_{l+1}), *and apply Algorithm* 4.1 *to solve the reduced problem and get approximate solutions $(\lambda_i, W\mathbf{x}_i)$, $i = 1, 2, \ldots, s$.*
4. *Set $X_{l+1} = W[\mathbf{x}_1\ \mathbf{x}_2\ \cdots \mathbf{x}_s]$.*

The block Rayleigh quotients Λ_l computed in step 1 need not always exist (see, e.g., Higham and Kim [8]). However, it is straightforward to check that Λ is a block Rayleigh quotient of (4.1) for X (i.e., it satisfies (5.1)) if it can be written in the form $\Lambda = Y\Omega Y^{-1}$, where $\Omega = \mathrm{diag}(\omega_i)$ is a matrix of eigenvalues and $Y = [\mathbf{y}_1\ \mathbf{y}_2\ \cdots\ \mathbf{y}_s]$ is a full rank matrix of eigenvectors for the associated reduced quadratic eigenvalue problem, i.e.,

$$(\omega_i^2 X^*\mathrm{M}X + \omega_i X^*\mathrm{C}X + X^*\mathrm{K}X)\mathbf{y}_i = 0, \quad i = 1, 2, \ldots, s. \tag{5.2}$$

Now the idea is to pick the first matrix X in Algorithm 5.1 in such a way that the block Rayleigh quotient exists. Suppose the approximate pairs $(\lambda_i, \mathbf{x}_i)$, $i = 1, 2, \ldots, s$, have been obtained by choosing s of the $2r$ approximate solutions resulting from solving a reduced problem of size r as in section 3.5. Let X be the matrix whose columns are exactly these approximate eigenvectors $\mathbf{x}_i$. Then the problem (5.2) has solutions $(\lambda_i, \mathbf{e}_i)$, $i = 1, 2, \ldots, s$, where $\mathbf{e}_i \in \mathbb{C}^s$ is the ith standard basis vector, so the matrix $\Lambda = \mathrm{diag}(\lambda_i)$ is in fact of the form $Y\Omega Y^{-1}$ with Y being the $s \times s$ identity matrix. Hence Λ is a block Rayleigh quotient of (4.1) for X. With these values for Λ and X, Algorithm 5.1 becomes the following.

ALGORITHM 5.2 (Block RQI 2). *This algorithm performs block Rayleigh quotient iterations to compute approximate eigenpairs for* (4.1), *starting only with eigenpairs obtained by solving a reduced quadratic problem.*
For $l = 1, 2, 3, \ldots$

0. *Given $(\lambda_i, \mathbf{x}_i)$, $i = 1, 2, \ldots, s$, a set of approximate eigenpairs for* (4.1) *obtained by solving a reduced problem as in section* 3.5.
1. *Solve ${\lambda_i}^2\mathrm{M}\mathbf{y}_{l+1,i} + \lambda_i\mathrm{C}\mathbf{y}_{l+1,i} + \mathrm{K}\mathbf{y}_{l+1,i} = \mathbf{x}_{l,i}$, $i = 1, 2, \ldots, s$, for the columns of Y_{l+1}. (Analogous to "shift-and-invert" in linear problems.)*

2. *Let W be a basis for* range(Y_{l+1})*, and apply Algorithm* 4.1 *to solve the reduced problem and get approximate solutions* $(\lambda_i, W\mathbf{x}_i)$*,* $i = 1, 2, \ldots, s$.

3. *Set* $X_{l+1} = W[\mathbf{x}_1\ \mathbf{x}_2\ \cdots \mathbf{x}_s]$.

The subspace approximation method switches to this RQI when the largest relative change in consecutive eigenvalue iterates remains less than some tolerance, i.e., when the eigenvalues have in a sense "nearly" converged.

Because of the form of the block equation solved in step 2 of Algorithm 5.1, the systems solved in step 1 of Algorithm 5.2 are in exactly the same form as the systems solved in Lancaster's original vector Rayleigh quotient iteration [13]. Other generalizations, such as solving $MY\Lambda^2 + CY\Lambda + KY = 2MX\Lambda + CX$, are possible, and potentially better, although we do not investigate them here.

5.2. Hybrid algorithm. Now we have discussed all the individual parts of the hybrid method and are ready to summarize the whole algorithm.

ALGORITHM 5.3. *Given nondefective eigenvalues* $\lambda_1, \lambda_2, \ldots, \lambda_l$ *of multiplicities* $n_1, n_2, \ldots n_l$*, and associated eigenvector matrices* $X_{(1)} \in \mathbb{C}^{n\times n_1}, X_{(2)} \in \mathbb{C}^{n\times n_2}, \ldots, X_{(l)} \in \mathbb{C}^{n\times n_l}$*, for* (2.1) *at* $t = 0$*, along with tolerances* tol_1 *(default* 10^{-3}*) and* tol*, this algorithm computes the corresponding values at* $t = 1$.

0. *Initialize:* relchange $= 2\text{tol}_1 \cdot \text{ones}(p, 1)$*;* $j = 0$*;* $W_0 = [\ ]$*;* convgct $= 0$*;*

If we know the left eigenvector matrix, anyleftvec $= 1$*, else* anyleftvec $= 0$.

1. *While* max(relerrs) $>$ tol *and* convgct < 2*,*

$j = j + 1$*;*

For $i = 1\colon l$*,* /* *compute new vectors, columns of* $X_{(i)}^{(j)}$*, to add to space* */

If anyleftvec $= 1$*, compute* $X_{(i)}^{(j)}$ *using Algorithm* 3.3*, else compute* $X_{(i)}^{(j)}$ *using Algorithm* 3.1.

Use modified Gram–Schmidt to compute basis W_j *for*

$$\mathcal{S}_j = \sum_{m=0}^{j} \left(\sum_{i=1}^{l} \text{range}(X_{(i)}^{(m)}) \right) = \left(\sum_{i=1}^{l} \text{range}(X_{(i)}^{(j)}) \right) + \text{range}(W_{j-1}).$$

Solve reduced problem in $\mathcal{S}_j$*; compute new* relerrs *and approximate* $(\lambda_i, \mathbf{x}_i)$ *using Algorithm* 4.1.

If $j > 1$ *set* relchange *to relative changes in computed eigenvalues from previous step.*

If max(relchange) $\le \text{tol}_1$*,* convgct $=$ convgct $+ 1$*; else* convgct $= 0$.

2. *While* max(relerrs) $>$ tol*,*

Apply Block RQI Algorithm 5.2*, beginning with* $X = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_p]$.

6. Complexity. The subspace approximation method spends over 95% of its time performing five tasks: computing the right-hand side V_k of (3.8) in the direct subspace computation; solving the least squares problems in the subspace computation or the linear systems in the block Rayleigh quotient iteration; applying modified Gram–Schmidt to compute bases; solving the reduced (projected) quadratic eigenvalue problems; and computing the error estimates.

For a crude analysis of these costs, assume that the maximum relative error estimate decreases by a factor ρ at each step. (Because the convergence is at least linear, such a value eventually exists.) If e_0 is the original maximum relative error estimate, the number of steps needed for convergence with tolerance tol is the smallest integer m greater than or equal to $\log_\rho \frac{e_0}{tol}$. Starting with p eigenpairs, the following are computed: pm values of V_k, requiring a total of at most $6pm$ matrix-vector products and $(m + 1)^2$ matrix-matrix products of size $p \times p$; pm least square solutions (or ill-conditioned linear system solutions); orthonormalization of $2p(m + 1)$ n-vectors;

TABLE 6.1
Work performed by the subspace approximation method.

Task	Flops
Computing V_k	$6(6pm(4nq) + (m+1)^2p^3)$
Linear solves (direct)	$6(pm)(4nq + 2nq^2)$
Modified Gram–Schmidt	$n(2p(m+1))^2$
Solving reduced problems	$\sum_{j=1}^{m+1}(8np^2j^2 + 4n^2jp + 200(jp)^3 + 8npj)$
Relative error estimates	$6(2p(m+1))(24nq + 6n)$
Total	$O(6q^2pnm + \frac{2}{3}p^2nm^3 + pn^2m^2 + 25p^3m^4)$

solutions of $m+1$ projected problems, with real bases of dimension $2p, 4p, \ldots, 2p(m+1)$, requiring work as shown in Table 3.1; and $2p(m+1)$ error estimates each requiring six matrix-vector products and three inner-products.

If $P(\lambda, t)$ is banded, let q denote the maximum of the upper and lower bandwidths of all the matrices in $P(\lambda, t)$. When the linear systems are solved directly, the method performs work at most on the order of that shown in Table 6.1, and, in fact, with the switch to the locally faster converging block RQI the total amount of work done should be even less. For general sparse matrices a similar analysis can be performed using the applicable flop counts for matrix-vector multiplications and the solution of linear systems or least squares problems.

7. Other methods. The subspace approximation approach described here is based on higher order perturbation analysis and knowledge of unperturbed solutions. It is interesting to look at connections with existing methods that either employ lower order perturbations or use no knowledge of previously computed eigenspaces. For example, for the problem $(\lambda^2\mathrm{I} - \lambda\mathrm{A} - \mathrm{B})\mathbf{x} = 0$, especially when A and B commute, some promising Krylov subspace methods are given by Hoffnung, Li, and Ye [10]. These methods build up the generalized Krylov subspaces $\mathcal{S}_j(\mathrm{A}, \mathrm{B}, \mathbf{q})$ one vector at a time, starting with some initial guess $\mathbf{q}$. If the matrix K in (1.1) is nonsingular, then, letting $\mu = \lambda^{-1}$, (1.1) can be rewritten as $(\mu^2\mathrm{I} + \mu\mathrm{K}^{-1}\mathrm{C} + \mathrm{K}^{-1}\mathrm{M})\mathbf{x} = 0$, and the subspaces generated in [10] are $\mathcal{S}_j(F\mathrm{M}, F\mathrm{C}, \mathbf{q})$, with $F = \mathrm{K}^{-1} = D(0)^{-1}$, where

$$D(\lambda) \equiv \lambda^2\mathrm{M} + \lambda\mathrm{C} + \mathrm{K}. \tag{7.1}$$

One example of a first order perturbation method is, of course, the vector Rayleigh quotient iteration, on which part of our method is based. In other first order perturbation methods (e.g., Ruhe [16]) the quadratic eigenvalue problem is approximated by a sequence of linear eigenvalue problems. A natural approach is to use the first order Taylor approximation $D(\lambda - \theta) \approx D(\lambda) - \theta D'(\lambda)$. At each step, for each eigenvalue estimate λ, we can solve the linear eigenvalue problem

$$D(\lambda)\mathbf{x} = \theta D'(\lambda)\mathbf{x}, \tag{7.2}$$

where θ is, for example, the eigenvalue of smallest magnitude. The associated eigenvector may be added to the subspace, and a reduced quadratic problem may then be solved to get the new λ values (or the new values may be taken as just $\lambda - \theta$). Clearly this approach avoids the problem of exponential subspace growth. We see that typically the intermediate problems here are n-dimensional complex generalized eigenvalue problems, as opposed to the least squares problems solved in the subspace approximation method. Thus the approach is suitable when $D(\lambda)$ and $D'(\lambda)$ are such that we have a good solver for problem (7.2).

For computing a single eigenvalue of a large, sparse, quadratic eigenvalue problem, the Jacobi–Davidson algorithm has been effective. The idea, discussed in Sleijpen et al. [17], is to build up a search space by (approximately) solving correction equations of the form

$$\left(\mathrm{I} - \frac{D'(\theta)\mathbf{u}\mathbf{u}^*}{\mathbf{u}^* D'(\theta)\mathbf{u}}\right) D(\theta)(\mathrm{I} - \mathbf{u}\mathbf{u}^*)\mathbf{t} = -D(\theta)\mathbf{u}, \tag{7.3}$$

where D is given by (7.1), $(\theta, \mathbf{u})$ is an approximation to the desired eigenpair, and the solution $\mathbf{t}$ is added to the search space using modified Gram–Schmidt. A reduced problem is solved in the search space to obtain the next approximate eigenpair. This method has been shown to be asymptotically quadratically convergent when (7.3) is solved exactly. Clearly $D(\theta)\mathbf{t} \in \mathrm{range}([D(\theta)\mathbf{u}\ D'(\theta)\mathbf{u}])$, so that the jth space, starting with the pair $(\theta, \mathbf{u})$, is contained in the generalized Krylov subspace $\mathcal{S}_j(F\mathrm{M}, F\mathrm{C}, \mathbf{u})$ with $F = D(\theta)^{-1}$. The connections among the related methods are compelling and subtle and should be studied further in the future.

8. Numerical examples.

8.1. A truss problem. Consider a long and slender truss structure shown in Figure 8.1. This example is designed to measure the effectiveness of the numerical algorithms for problems with proportional and nonproportional damping.

The element stiffness and mass matrices are given as

$$K_e = \frac{AE}{l}\begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad M_e = \frac{\rho A l}{6}\begin{pmatrix} 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \end{pmatrix}.$$

For each member let the cross-sectional area A be 1.0 and assume the other constants have the values $E = 10^7$ and $\rho = 1.0$. Values of the length l are given in the figure. The assembled matrices K and M are symmetric positive definite of order 2000. The members are numbered as shown, so that K has a bandwidth of 13 and M has a bandwidth of 12. Assume the eight smallest eigenpairs $(\mu_1, \mathbf{x}_1), \ldots, (\mu_8, \mathbf{x}_8)$ for the generalized eigenvalue problem $\mathrm{K}\mathbf{x} = \mu\mathrm{M}\mathbf{x}$ have been computed, i.e., assume the undamped problem has been solved.

8.1.1. Proportional damping. The first example is of simple proportional damping, with 5% damping in the first eight modes and zero damping in all the others. The damping matrix is then $\mathrm{C}_{\mathrm{prop}} = \mathrm{M}(\sum_{j=1}^{8} 2(.05)\omega_j \phi_j \phi_j^T)\mathrm{M}$, where $\omega_j = \sqrt{\mu_j}$ and $\phi_j = (\mathbf{x}_j^T \mathrm{M}\mathbf{x}_j)^{-1/2}\mathbf{x}_j$ for $j = 1, \ldots 8$. This is a symmetric positive semidefinite matrix

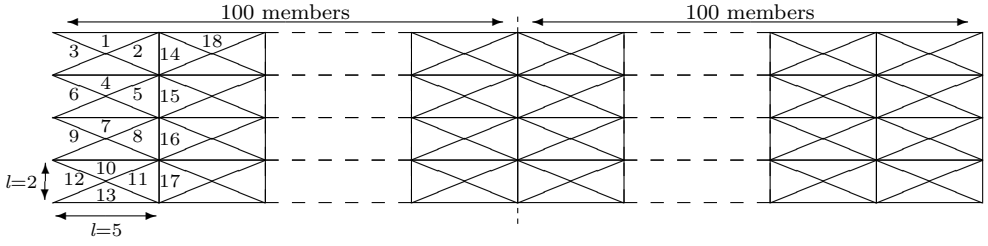


FIG. 8.1. *Truss structure.*

TABLE 8.1
Comparison of methods for proportionally damped truss problem.

	Subsp. Approx.	RQI	**eigs**
CPU time (seconds)	0.35	25.4	57
max *relerrs* (estimate)	6.56×10^{-6}	8.47×10^{-6}	9.92×10^{-6}
max *rel.err.* (actual)	1.52×10^{-8}	1.97×10^{-8}	1.36×10^{-8}

of rank 8, which is dense but easy to apply as an operator. Taking the undamped problem $(\lambda^2 \mathrm{M} + \mathrm{K})\mathbf{x} = 0$ as the original, unperturbed problem, to which solutions are known, we use the subspace approximation method to solve the damped problem $(\lambda^2 \mathrm{M} + \lambda \mathrm{C}_{\mathrm{prop}} + \mathrm{K})\mathbf{x} = 0$ for the 16 eigenvalues of least magnitude. As shown in section 3.4, the subspace computation involves only real arithmetic.

The true eigenvalues of the damped equation are simply the roots of the quadratic polynomials $\lambda^2 + 0.1\lambda\sqrt{\mu_j} + \mu_j$ and, as expected, the subspace approximation method computes these values in the first step, using the first subspace, since the original eigenspaces and final eigenspaces are the same. The acceptance tolerance is taken to be 10^{-5}. The first eight pairs of paths are nearly linear and all other eigenvalue paths are constant; there is no risk of path crossing. Table 8.1 shows a comparison with vector RQI started from the same original values, and with **eigs**, the MATLAB implementation of ARPACK (see [15]), applied to the second linearization of (1.2) with $\mathrm{N} = \mathrm{I}$. All three methods converged to the correct solutions, but the subspace approximation method is clearly appropriate for this problem, and one sees that the method's performance is orders of magnitude better than that of the other two methods.

8.1.2. Nonproportional damping. In the second example, half the structure has 1% damping and the other half has 2% damping (to the right and left, resp., of the dotted vertical center line in Figure 8.1). The resulting damping matrix $\mathrm{C}_{\mathrm{npr}}$, assembled as indicated in Figure 8.2, is composed of two overlapping submatrices along the diagonal, each of which is a different linear combination of the corresponding submatrices of M and K associated with the two different damping percentages. The values a_{ij} used in the linear combinations are given by $\binom{a_{i1}}{a_{i2}} = \frac{2\xi_i}{\omega_1+\omega_2}\binom{\omega_1\omega_2}{1}$, where $\xi_1 = 0.01$, $\xi_2 = 0.02$, and $\omega_1 = \sqrt{\mu_1}$ and $\omega_2 = \sqrt{\mu_2}$ are the two smallest natural frequencies for the undamped problem. Because of the elements in common between the 1% and 2% damping, $\mathrm{C}_{\mathrm{npr}}$ is not itself a combination of M and K of the form $\sum_{j=0}^{p} \alpha_j \mathrm{M}(\mathrm{M}^{-1}\mathrm{K})^j$, so this is nonproportional damping (see, e.g., Clough and Penzien [3, Chapter 12]) and cannot be solved using the traditional modal superposition.

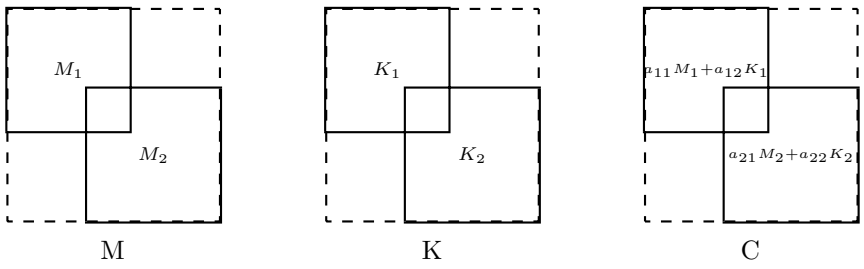


FIG. 8.2. *Block form of nonproportional damping matrix.*

TABLE 8.2
Comparison of methods for nonproportionally damped truss problem.

	Subsp. Approx.	RQI	**eigs**
CPU time (sec.)	1.85	6.58	11.31
Maximum *relerrs*	8.75×10^{-7}	9.80×10^{-7}	7.25×10^{-7}
Max. $\angle$ error (rad.)	8.02×10^{-8}	8.82×10^{-8}	8.30×10^{-8}

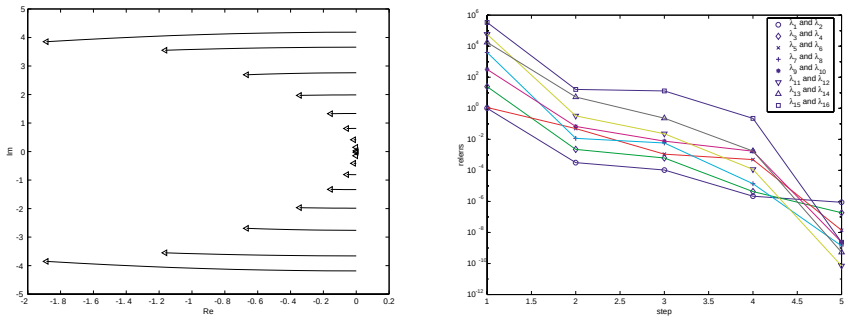


FIG. 8.3. *Eigenvalue paths and convergence of subspace approximation method for nonproportional problem.*

The results of applying the subspace approximation method to compute the first 16 eigenvalues of the damped problem are summarized in Table 8.2, again compared with RQI and **eigs**. The acceptance tolerance is taken to be 10^{-6}. In this problem, iterative solution via QMR with MATLAB's default parameters is used both in the subspace approximation and in the vector RQI. (The Jacobi–Davidson method was also tried here using QMR, but it required over 383 seconds to achieve the desired accuracy, possibly because a more suitable linear solver is needed. Solving the quadratic eigenvalue problem via a sequence of linear approximations was tried, using a symmetric indefinite Lanczos solver for the inner problem, but this approach did not converge, again probably because a more suitable inner solver is required.) Linearization ((1.2), first equation) is used in **eigs**, with the choice N = K and with the default parameters because no advantage was found in making other choices. One sees that the relative accuracy of the computed eigenvalues is about the same for all three methods, as is the order of the error angles in the computed eigenvectors. Figure 8.3 shows the eigenvalue paths (left graph) and convergence of the subspace approximation method (right graph). The paths are less linear than those for the proportional damping problem, but they are smooth and well separated from other eigenvalue paths, showing that the problem is very suitable for subspace approximation. Note that faster convergence could have been achieved by switching to the block RQI early, after step 2, yielding a time savings of 8.6%. However, further work is needed to automate the "tweaking" of the hybrid method.

8.2. Humboldt Bay Middle Channel Bridge example. This example illustrates the possible application of the subspace approximation method in the analysis of a bridge structure including soil properties. The following is a description from Conte et al. [4] of the Humboldt Bay Middle Channel Bridge:

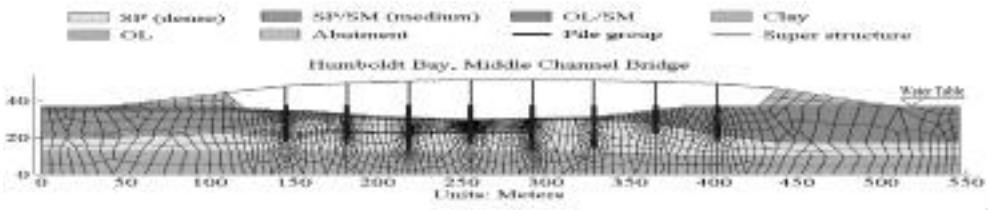


FIG. 8.4. *Finite element model of Middle Channel Bridge.*

TABLE 8.3
Mass, stiffness, and damping matrices for the bridge problem.

Matrix	Bandedness	Sparsity	max(eig)	min(eig)
M	diagonal	5038 nonzero elts.	1.31×10^3	0
K	bandwidth 5102	0.3% nonzero	4.94×10^{11}	9.95×10^2
C_2	bandwidth 5102	0.3% nonzero	1.12×10^9	2.26×10^0
C_7	bandwidth 5102	0.3% nonzero	1.12×10^9	7.93×10^0

> The Humboldt Bay Middle Channel Bridge, near Eureka in northern California . . . , is a 330 meters long, 9-span composite structure . . . . It is supported on eight pile groups, each of which consists of 5 to 16 prestressed concrete piles, in soils vulnerable to liquefaction (under extreme earthquake shaking conditions). The river channel has an average slope from the banks to the center of about 7% (4 degrees). The foundation soil is composed of mainly dense fine-to-medium sand (SP/SM), organic silt (OL), and/or stiff clay layers. In addition, thin layers of loose sand and soft clay (OL/SM) are located near the ground surface . . . . A two-dimensional nonlinear model of the Middle Channel Bridge, including the superstructure, piers, and supporting piles was developed . . . as shown in [Figure 8.4].

The example here is not intended to accurately model the various soil properties since each soil type could have its own frequency dependent damping properties. Nevertheless, for illustrative purposes, the following realistic damping values are tested: 2% damping on the bridge structure, and first 2%, then 7% damping on the soil (denoted here, respectively, as C_2 (a proportional damping matrix) and C_7 (a nonproportional damping matrix)). The properties of the 5164×5164 symmetric matrices M, C, and K are given in Table 8.3.

Since M is singular, with 126 zero diagonal elements, and K is positive definite, to guarantee continuous eigenvalue paths we may swap the roles of M and K and instead solve for the largest eigenvalues μ for the problems

$$(\mu^2 \widehat{\mathrm{K}} + \widehat{\mathrm{M}})\mathbf{x} = 0 \quad \text{(undamped)}, \tag{8.1}$$

$$(\mu^2 \widehat{\mathrm{K}} + \mu \widehat{\mathrm{C}} + \widehat{\mathrm{M}})\mathbf{x} = 0 \quad \text{(damped)}. \tag{8.2}$$

The desired eigenvalues are then the reciprocals $\lambda = \frac{1}{\mu}$, and the eigenvectors are unaffected by the interchange. This interchange is not needed for the subspace approximation method since we are interested only in the smallest (finite) eigenvalue paths. However, when the corresponding linearized problem ((1.2), second equation) is solved for comparison purposes, the interchange is necessary in order to have a symmetric positive definite matrix on the right-hand side of the linearized eigenvalue problem, $\mathcal{A}\mathbf{x} = \lambda \mathcal{B}\mathbf{x}$.

TABLE 8.4
CPU times for bridge example.

Problem	Method	CPU time (sec.)
Undamped	**eigs**	4.7
Damping: C_2	Subsp. Approx. Starting from $C = 0$	0.9
Damping: C_7	Subsp. Approx. starting from C_2	58
Damping: C_7	Subsp. Approx. starting from $C = 0$	38
Damping: C_7	**eigs**, using linearization (1.2), second equation	159

We begin by using the MATLAB **eigs** function to compute the first 20 eigenpairs of the undamped problem. Taking these as the unperturbed solutions, the subspace approximation method is then applied to solve the damped problems, with the MATLAB "slash" operator to solve the least squares problems. The CPU times for these results are shown in Table 8.4. Solving the undamped problem using **eigs** and then solving the problem with damping matrix C_7 using the subspace approximation method takes 44 seconds of CPU time, and even when an intermediate problem with damping matrix C_2 is computed, the total time required is 64 seconds, 40% of the time required by **eigs** to solve the linearized problem. (When started with the undamped problem, the subspace computation involves only real arithmetic, so less work is performed in that case.) If the undamped problem has been solved previously and its solutions are already available, using the subspace approximation method to compute the solutions with damping matrix C_7 requires 25% of the time required by **eigs** to solve the same problem.

8.3. Path crossing example. Unlike the previous examples, the problem in this section displays changes in eigenvalue order as well as some switching from complex to real values along the eigenvalue paths.

In this example, M and K are given as BCSSTM12 and BCSSTK12 from the Harwell–Boeing collection (see Duff, Grimes, and Lewis [5]). These matrices have order 1473, and the matrix C is taken to be the block combination of M and K such that if $M_1 = M(1\colon 600, 1\colon 600)$ and $M_2 = M(540\colon 1473, 540\colon 1473)$, and if K_1, K_2 are defined in the same way from K, then

$$c_{ij} = \begin{cases} a_{11}m_{ij} + a_{12}k_{ij}, & \text{when } i < 540 \text{ or } j < 540, \\ (a_{11} + a_{21})m_{ij} + (a_{12} + a_{22})k_{ij}, & \text{when } 540 \le i,\ j \le 600, \\ a_{21}m_{ij} + a_{22}k_{ij}, & \text{when } i > 600 \text{ or } j > 600, \end{cases}$$

where $\binom{a_{i1}}{a_{i2}} = \frac{2\xi_i}{\omega_1+\omega_2}\binom{\omega_1\omega_2}{1}$ with $\xi_1 = 0.05$, $\xi_2 = 0.10$, and ω_1 and ω_2 as the first and tenth natural frequencies for the undamped problem. Now $\|M\| = 1.34 \times 10^1$, $\|C\| = 6.68 \times 10^5$, and $\|K\| = 6.56 \times 10^8$. The eigenvalue paths are shown in Figure 8.5. Path crossing and order changes can be observed. The subspace approximation method correctly computes, with tolerance 10^{-5}, the first 20 perturbed eigenvalues and vectors starting with the first 10 complex conjugate pairs of eigenvalues and corresponding eigenvectors. The vector RQI method, started with the same values and using the same convergence tolerance, computes only the values numbered 4, 5, 6, 7, 12, 13, 14,

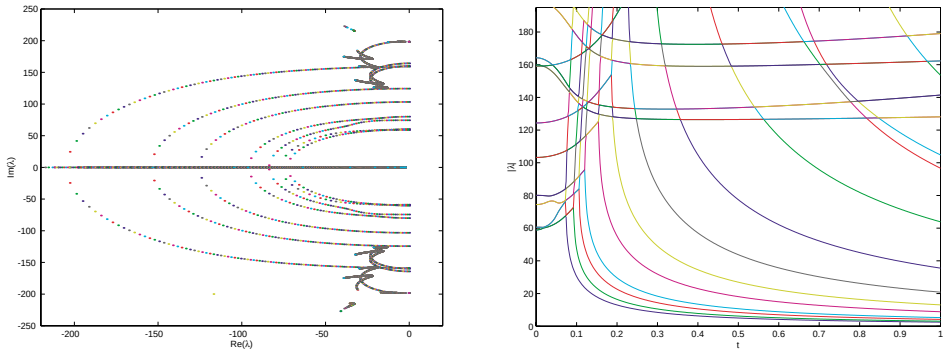


FIG. 8.5. *Eigenvalue paths and moduli of eigenvalues for size* 1473 *example.*

TABLE 8.5
Results for size 1473 *example.*

	Subsp. Approx.	**eigs**
Maximusm relerrs	2.5×10^{-7}	1.1×10^{-3}
Maximum $\angle$ error est. (rad.)	3.7×10^{-6}	1.7×10^{-3}
CPU time	92	77

15, 17, and 18 (ordered by magnitude). **Eigs** computes all 20 values and takes only 77 seconds of CPU time versus 92 using the subspace approximation method. However the solutions from **eigs** are much less accurate, as can be seen in Table 8.5, and decreasing the tol parameter of **eigs** by a factor of 10^5 causes negligible improvement in the errors while increasing the computation time to 124 seconds.

9. Summary and future directions. In this paper we developed a method for computing a few eigenvalues and eigenvectors of a quadratic eigenvalue problem assuming that solutions to the corresponding generalized (undamped) eigenvalue problem are known. The Taylor series for the block eigenvector matrix $X(t)$ was shown to converge, and the range of the kth Taylor polynomial was shown to be contained in the kth generalized Krylov subspace $\mathcal{S}_k(F\mathrm{M}, F\Delta\mathrm{C}, X_0)$, where F is a matrix such that $FP(\lambda_0, 0)$ fixes a space complementary to the range of the original block eigenvector matrix X_0.

We discussed how to compute the generalized Krylov subspaces by solving a sequence of least squares problems, and also how to directly compute the derivative subspaces range$([X_0\ X^{(1)}\ \cdots\ X^{(k)}])$ assuming that certain additional assumptions hold. Computing reduced problems in these subspaces was described. Using a first order error analysis a reasonable acceptance criterion was developed. After generalizing Lancaster's Rayleigh quotient iteration to a block algorithm, we assembled a hybrid method starting with the linearly converging subspace approximations and switching to the faster converging RQI as a finishing procedure. From several numerical experiments it is clear that solving the quadratic eigenvalue problem as a perturbed quadratic eigenvalue problem using the subspace approximation method has some advantages both in speed and in accuracy over solving the problem from scratch using a standard linearization approach.

The theory in this paper extends to more general perturbed quadratic eigenvalue problems and to other polynomial eigenvalue problems [11]. Suppose that

$(\lambda^N(t)\widehat{\mathrm{A}}_N(t)+\lambda^{N-1}(t)\widehat{\mathrm{A}}_{N-1}(t)+\cdots+\lambda(t)\widehat{\mathrm{A}}_1(t)+\widehat{\mathrm{A}}_0(t))\mathbf{x}(t)=0$ and that $FP(\lambda_0,0)$ fixes a space complementary to range(X_0). Then range$(\widehat{X}^{(j)}) \subseteq \mathcal{S}_j(F\mathrm{A}_1,\ldots,F\mathrm{A}_N, F\Delta\mathrm{A}_0,F\Delta\mathrm{A}_1,\ldots,F\Delta\mathrm{A}_N,X_0)$ for $j=0,1,2,\ldots$. Future study of such extensions should prove fruitful.

Another important direction for future work is toward the reduction of the subspace dimension. Suppose a dense standard or generalized eigenvalue problem of size up to N is considered small and a problem of size greater than N is considered large. Then we should ensure that the reduced problems our method requires to be solved remain "small." The derivative subspaces grow linearly, while the generalized Krylov subspaces $\mathcal{S}_j$ can grow exponentially with j; either way a mechanism is needed for stopping the growth when the subspace reaches size $N/2$.

Three possible approaches are (1) switching early to block RQI; (2) restarting with a subspace of dimension s spanned by the approximate eigenvectors based on the fact that the approximate eigenpairs $\{(\mu_i,\mathbf{y}_i)\}$ are exact solutions to the problems

$$\left(\lambda^2\widehat{\mathrm{M}}(1)+\lambda\widehat{\mathrm{C}}(1)+\widehat{\mathrm{K}}(1)-\frac{\mathbf{r}_i\mathbf{u}_i^T}{\mathbf{u}_i^T\mathbf{y}_i}\right)\mathbf{x}=0,$$

where $\mathbf{u}_i^T\mathbf{y}_i\neq 0$, $i=1,2,\ldots,s$; and (3) cutting the timestep and using a homotopy continuation method. (This approach is also appropriate for handling larger perturbations, when the convergence radius of the Taylor series for the eigenvector matrix $X(t)$ is less than 1.)

In summary, there are several interesting avenues to pursue in continuing this work on the subspace approximation method for perturbed, quadratic, and polynomial eigenvalue problems.

REFERENCES

[1] L. V. Ahlfors, *Complex Analysis*, 3rd ed., McGraw–Hill, New York, 1979.

[2] O. A. Bauchau, *A solution of the eigenproblem for undamped gyroscopic systems with the Lanczos algorithm*, Internat. J. Numer. Methods Engrg., 23 (1986), pp. 1705–1713.

[3] R. W. Clough and J. Penzien, *Dynamics of Structures*, 2nd ed., McGraw–Hill, New York, 1993.

[4] J. P. Conte, A. Elgamal, Z. Yang, Y. Zhang, G. Acero, and F. Seible, *Nonlinear seismic analysis of a bridge ground system*, in Proceedings of the 15th ASCE Engineering Mechanics Conference, Columbia University, New York, 2002, CD-ROM, ASCE, Reston, VA, 2002.

[5] I. S. Duff, R. G. Grimes, and J. G. Lewis, *Users' Guide for the Harwell–Boeing Sparse Matrix Collection (Release* I*)*, Technical report RAL 92-086, Chilton, Oxon, England, 1992.

[6] R. J. Duffin, *A minimax theory for overdamped networks*, J. Rational Mech. Anal., 4 (1955), pp. 221–233.

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

[8] N. J. Higham and H.-M. Kim, *Numerical analysis of a quadratic matrix equation*, IMA J. Numer. Anal., 20 (2000), pp. 499–519.

[9] M. E. Hochstenbach and H. A. van der Vorst, *Alternatives to the Rayleigh quotient for the quadratic eigenvalue problem*, SIAM J. Sci. Comput., 25 (2003), pp. 591–603.

[10] L. Hoffnung, R.-C. Li, and Q. Ye, *Krylov Type Subspace Methods for Matrix Polynomials*, Technical report 2002-08, Department of Mathematics, University of Kentucky, Lexington, KY, 2002.

[11] U. B. Holz, *Subspace Approximation Methods for Perturbed Quadratic Eigenvalue Problems*, Ph.D. thesis, Stanford University, Stanford, CA, 2002.

[12] T. Kato, *Perturbation Theory for Linear Operators*, corrected printing of the 2nd ed., Springer-Verlag, Berlin, 1980.

[13] P. Lancaster, *A generalized Rayleigh quotient iteration for lambda-matrices*, Arch. Rational Mech. Anal., 8 (1961), pp. 309–322.
[14] P. Lancaster, *Lambda-Matrices and Vibrating Systems*, Pergamon Press, Oxford, UK, 1966.
[15] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, Software Environ. Tools 6, SIAM, Philadelphia, 1998.
[16] A. Ruhe, *A rational Krylov algorithm for nonlinear matrix eigenvalue problems*, Zap. Nauchn. Sem. S.-Peterburg. Otdel. Mat. Inst. Steklov. (POMI), 268 (2000), pp. 176–180.
[17] 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.
[18] F. Tisseur, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra Appl., 309 (2000), pp. 339–361.
[19] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.
[20] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.
[21] T. Zhang, G. H. Golub, and K. H. Law, *Eigenvalue perturbation and generalized Krylov subspace method*, Appl. Numer. Math., 27 (1998), pp. 185–202.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 2, pp. 522–543

IS JACOBI–DAVIDSON FASTER THAN DAVIDSON?*

YVAN NOTAY†

Abstract. The Davidson method is a popular technique to compute a few of the smallest (or largest) eigenvalues of a large sparse real symmetric matrix. It is effective when the matrix is nearly diagonal, that is, when the matrix of eigenvectors is close to the identity matrix. However, its convergence properties are not yet well understood, and neither is how it behaves compared to the more recent Jacobi–Davidson method, for which a proper convergence analysis exists. In this paper, we develop a new convergence analysis of the Davidson method. This analysis proves that the convergence is fast for nearly diagonal matrices when the method is initialized in the standard way. One may at this stage not expect any significant improvement by shifting to the Jacobi–Davidson method. On the other hand, the latter may be more effective for more general initial approximations. It is also best suited for matrices that are not nearly diagonal, thanks to the use of more sophisticated preconditioning and/or inner iterations.

Key words. eigenvalue, Davidson, Jacobi–Davidson, preconditioning, symmetric matrices

AMS subject classification. 65F15

DOI. 10.1137/S0895479803430941

1. Introduction.

For the computation of a few extreme eigenpairs of a symmetric matrix, the Davidson method [3] has been appreciated since its introduction by many application scientists working in fields such as quantum chemistry or quantum physics. In this context, the matrix is often nearly diagonal, and this has been observed over the years to favor a rapid convergence. However, this observation has, up to now, not been confirmed by proper theoretical results. More surprisingly, available analyses even lead to fear that the method could stagnate when the matrix is too close to a diagonal one [2, 23].

From that point of view, the Jacobi–Davidson (JD) method [23] offers a proper enhancement since this method tends to be very close to the Rayleigh quotient iteration [18] when the preconditioner tends to be exact, as it occurs when using a diagonal preconditioning for a matrix that is very close to a diagonal one. Moreover, recent analyses confirm this claim and, more generally, prove that the better the preconditioner, the faster the convergence [14, 16, 21, 27].

Now, the JD method has mainly gained popularity in situations for which the Davidson method would not fit well anyway, such as eigenvalue computation involving unsymmetric matrices and/or some interior part of the spectrum. Indeed, to the best of our knowledge, the above-mentioned potential weakness of the Davidson method has never been reported in practical situations. More generally, considering cases for which the method works reasonably well, little is known about which benefit could be obtained by using the more sophisticated JD approach.

In this paper, we investigate this question. The Davidson and JD methods are described in section 2. They are first numerically compared on a small but instructive example in section 3. Then, the convergence of the Davidson method is analyzed in section 3, where we also develop a theoretical comparison with the JD method. These

*Received by the editors July 2, 2003; accepted for publication (in revised form) by H. van der Vorst April 8, 2004; published electronically January 12, 2005. This research was supported by the "Fonds National de la Recherche Scientifique," Maître de recherches.

http://www.siam.org/journals/simax/26-2/43094.html

†Service de Métrologie Nucléaire (C.P. 165/84), Université Libre de Bruxelles, 50, Av. F.D. Roosevelt, B-1050 Brussels, Belgium (ynotay@ulb.ac.be).

results are then illustrated in section 4 by some further (more realistic) numerical experiments.

Notation. Throughout this paper, A is a real symmetric $n \times n$ matrix. (The extension to complex Hermitian matrices is straightforward, but we confine ourselves to the real case for sake of simplicity.) The eigenpairs are denoted $(\lambda_i, \mathbf{x}_i)$, $i = 1, \ldots, n$ with the eigenvalues ordered increasingly (i.e., $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$) and the eigenvectors orthonormal (i.e., $(\mathbf{x}_i, \mathbf{x}_j) = \delta_{ij}$).

For any $n \times j$ matrix V, span(V) is the subspace spanned by the columns of V. We also denote $\mathbf{e}_j$ the canonical vector defined by $(\mathbf{e}_j)_i = \delta_{ij}$.

2. Davidson and JD methods. Both methods obey to the same general template, which we give in Algorithm 2.1. (More precisely, we give the version that is appropriate to the case where one searches for the smallest eigenvalue of a symmetric matrix.)

ALGORITHM 2.1 (Davidson/Jacobi–Davidson).

Choose an initial subspace:
Select a set of k_0 (≥ 1) *orthonormal vectors* $\mathbf{v}_i$, $i = 1, \ldots, k_0$
and set $V_1 = [\mathbf{v}_1 \cdots \mathbf{v}_{k_0}]$.
For $k = 1, \ldots$ **do**
1. *Compute the interaction matrix* $H_k = V_k^T A V_k$.
2. *Compute the smallest eigenpair* (θ_k, y_k) *of* H_k *(with* $\|y_k\| = 1$*).*
3. *Compute the Ritz vector* $\mathbf{u}_k = V_k y_k$.
4. *Compute the residual* $\mathbf{r}_k = (A - \theta_k I)\mathbf{u}_k$.
5. **if** *convergence* **then** *exit*
6. *Shrink the search space if needed:*
if $\dim(\text{span}(V_k)) \geq j_{\max}$ **then**
compute the $j_{\min}$ *smallest eigenpairs* $(\theta_k^{(j)}, y_k^{(j)})$ *of* H_k*;*
compute the Ritz vectors $\mathbf{u}_k^{(j)} = V_k y_k^{(j)}$*;*
reset $V_k = [\mathbf{u}_k^{(1)} \cdots \mathbf{u}_k^{(j_{\min})}]$.
7. *Compute new direction* $\mathbf{t}_k$ *from* $\mathbf{r}_k$.
8. *Orthonormalize* $[V_k \ \mathbf{t}_k]$ *into* V_{k+1}.

End For
$j_{\min}, j_{\max}$*: integer parameters such that* $0 < j_{\min} < j_{\max}$

Thus, both methods search for the smallest eigenpair in a subspace of increasing dimension, which is shrunk from time to time to avoid excessive memory requirements and numerical cost per iteration. They differ in substep 7, by the way $\mathbf{t}_k$ is computed.

The Davidson method uses

$$\mathbf{t}_k^{(D)} \ = \ M_k^{-1}\mathbf{r}_k, \tag{2.1}$$

where M_k is an approximation to $A - \theta_k I$ which is given by

$$M_k \ = \ \text{diag}(A) - \theta_k I \tag{2.2}$$

when one uses the method in its standard settings.

Instead, the JD method computes $\mathbf{t}_k$ by solving (in general, approximately) the so-called correction equation

$$\big(I - \mathbf{u}_k\mathbf{u}_k^T\big)\big(A - \theta_k I\big)\big(I - \mathbf{u}_k\mathbf{u}_k^T\big)\mathbf{t} = -\mathbf{r}_k; \quad \mathbf{t} \perp \mathbf{u}_k. \tag{2.3}$$

Usually, this is done by running a few steps of a Krylov subspace iterative solver with preconditioning

$$\widetilde{M}_k = \left(I - \mathbf{u}_k \mathbf{u}_k^T\right) M_k \left(I - \mathbf{u}_k \mathbf{u}_k^T\right), \tag{2.4}$$

M_k being here also some approximation to $A - \theta_k I$. A scheme relatively close to the Davidson method is obtained when one skips inner iterations and performs a single application of the preconditioner (2.4). Indeed, the solution to

$$\widetilde{M}_k \, \mathbf{t} = -\mathbf{r} \; ; \quad \mathbf{t} \perp \mathbf{u}_k \tag{2.5}$$

is (see [23])

$$\mathbf{t}_k^{(JD)} = -M_k^{-1} \, \mathbf{r}_k + \frac{(\mathbf{u}_k, \, M_k^{-1} \, \mathbf{r}_k)}{(\mathbf{u}_k, \, M_k^{-1} \, \mathbf{u}_k)} \, M^{-1} \, \mathbf{u}_k. \tag{2.6}$$

Both methods start then with the same vector $\mathbf{t}_k^{(D)}$, but JD adds an oblique projection onto $\mathbf{u}_k^{\perp}$. (Note that substep 8 of Algorithm 2.1 also implies a projection onto $\mathbf{u}_k^{\perp}$, but the latter is orthogonal).

Whenever using the preconditioner (2.2), a standard way to initialize the Davidson method (the default strategy in [26]) consists of selecting a few canonical vectors, more precisely those corresponding to the k_0 smallest diagonal elements. Of course, if A is nearly diagonal, this is a relevant choice for the JD method, too. Letting a_{ii} be the smallest diagonal element, this strategy ensures $\theta_1 < a_{ii}$ as soon as the starting basis contains, besides $\mathbf{e}_i$, one more canonical vector $\mathbf{e}_j$ corresponding to an index j for which $a_{ij} \neq 0$ [2]. Since θ_k forms a nonincreasing sequence [18], it means that the diagonal preconditioner (2.2) is positive definite for all k. Note that the convergence of the Davidson method is guaranteed when M_k is positive definite for all k [2, Theorem 2.1]. However, little is known about the convergence speed.

"Generalized" Davidson methods [2, 8] have been proposed that make use of

$$M_k = G - \theta_k I \tag{2.7}$$

instead of (2.2), where G stands for some closer approximation to A, e.g., its tridiagonal part (the factorization of M_k has to remain cheap). Observe here that any analysis developed for the case (2.2) is easily extended to the general case by considering a basis transformation that makes G diagonal. However, since one does not know the canonical vectors in this transformed basis (actually the eigenvectors of G), one cannot in this case apply the above strategy to obtain a nice starting subspace with M_k positive definite from the beginning. As will be seen below, this may have practical consequences.

Finally, both Davidson and JD approaches can be used with a constant, θ_k-independent preconditioner. Actually, this is the standard choice for the JD method, for which one usually selects M_k equal to some approximation of $A - \tau I$ for some "target" τ [4, 24]. Concerning the Davidson update (2.1), note, however, that if M_k is a positive definite matrix that does not depend on k anymore, then the method becomes an (subspace accelerated) inexact inverse iteration scheme, for which proper analyses already exist [6, 10, 11, 15]. Moreover, the locally optimal block preconditioned conjugate gradient (LOBPCG) method from [5] then seems to be more efficient than standard subspace acceleration, being close to optimal without need for restarting. Therefore, in what follows, we only consider the standard Davidson method defined

by (2.1) or (2.2) (or (2.7)). We refer to [13, 14] for some comparison between inexact inverse iteration, the LOBPCG and JD methods, and to [7] for a wider survey of preconditioned eigensolvers.

3. A small illustrative experiment. Qualitative convergence analyses of the Davidson method have been developed in [2, 8]. Quantitative results seem more difficult to obtain. Probably connected to this fact is the observation that if M_k converges to $A - \theta_k I$, then

$$\mathbf{t}_k^{(D)} = M_k^{-1}\mathbf{r}_k = M_k^{-1}(A - \theta_k\, I)\, \mathbf{u}_k$$

should converge to $\mathbf{u}_k$. Hence, either the current subspace is not expanded, or it is expanded in a random fashion by rounding errors. However, besides some artificial experiments as those reported in [23], poor behavior of the method because of too good preconditioning does not seem to correspond to the actual practice. We explain this by the following observation: M_k converging to $A-\theta_k\, I$ in the usual sense implies that, at some stage, M_k becomes indefinite (since $A - \theta_k\, I$ itself is indefinite). Therefore, what may then happen does not necessarily correspond to what is observed when one takes care to preserve the positive definiteness of M_k. As we have already mentioned, the latter suffices to guarantee the convergence of the method [2, Theorem 2.1].

Before developing our mathematical analysis in the next section, we first illustrate these considerations by the following small experiment. The matrix is 10×10 and given by

$$a_{ij} = \begin{cases} i & \text{if } j = i, \\ \alpha & \text{otherwise,} \end{cases}$$

where α is a parameter. For both the Davidson method (defined by (2.1), (2.2)) and the JD method (defined by (2.2), (2.6)), we computed the exact "local" convergence rate

$$\sigma = \frac{(\theta_{k+1} - \lambda_1)}{(\theta_k - \lambda_1)} \tag{3.1}$$

when $V_k = [\mathbf{u}_k]$, that is, when the subspace has just been shrunk with $j_{\min} = 1$. (Note that θ_k is equal to the Rayleigh quotient corresponding to $\mathbf{u}_k$ and hence cannot be smaller than λ_1.)

Two situations were considered. In the first one,

$$\mathbf{u}_k = \sqrt{1 - s^2}\ \mathbf{x}_1 + s(I - \mathbf{x}_1\, \mathbf{x}_1^T)\, \mathbf{e},$$

where s, $-1 \le s \le 1$, is a parameter and $\mathbf{e} = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}^T$. In Figure 1, we have plotted σ against $|s|$ for both methods and two values of α. Note that in each case there are two curves, one for positive s and one for negative s. Dotted vertical lines have been added to indicate the values of $|s|$ corresponding to $\theta_k = a_{11}$ (i.e., $\theta_k = 1$) and $\theta_k = \lambda^* \equiv (\lambda_1 + \lambda_2)/2$. The first value corresponds indeed to the point from where the convergence of the Davidson method is guaranteed because M_k is positive definite, whereas the second value corresponds to the point from where the analysis in [14] applies to prove the convergence of the JD method; note that θ_k decreases with $|s|$, $\mathbf{u}_k$ becoming closer to $\mathbf{x}_1$.

One sees that the convergence of the Davidson method is rather unpredictable when M_k is not positive definite ($\theta_k \ge 1$), stagnation being possible. Beyond this

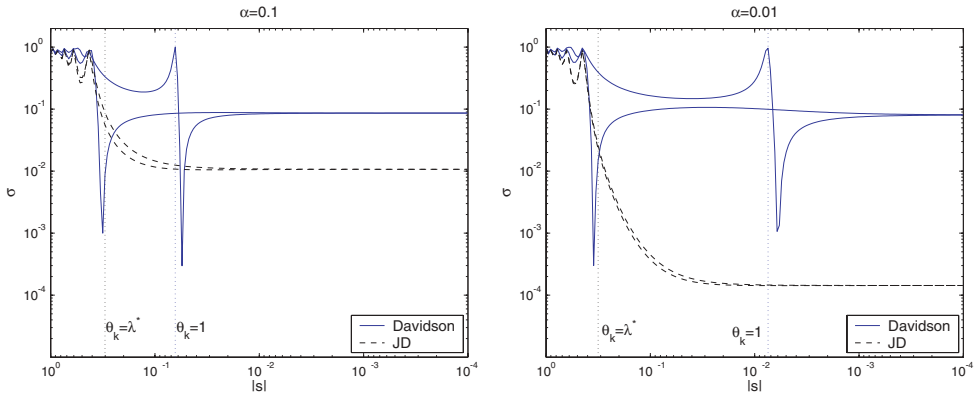


FIG. 1. *σ versus $|s|$ (first situation).*

limit, and as $|s|$ decreases, σ converges quickly to a value independent of α and close to 10^{-1}. On the other hand, the JD method is stable as soon as $\theta_k < \lambda^*$ with the asymptotic convergence rate approximately equal to 10^{-2} for $\alpha = 10^{-1}$ and 10^{-4} for $\alpha = 10^{-2}$.

Now, this example does not represent truly what happens when the Davidson method is initialized in the standard way. Indeed, by including the first canonical vector into V_{k_0}, one not only ensures the positive definiteness of M_k, but one also enforces the condition $(\mathbf{r}_k)_1 = 0$ (since, as a result of the Ritz–Galerkin process used to compute $\mathbf{u}_k$, the residual vector is orthogonal to any vector in V_k). To figure why this may have a significant influence on the computation, observe that $\mathbf{e}_1$ is precisely the eigenvector of M_k corresponding to its smallest eigenvalue—the one which makes M_k ill conditioned. (When the matrix is close to a diagonal one, $\theta_k \approx \lambda_1$ implies $\theta_k \approx a_{11}$.) By making $\mathbf{r}_k$ orthogonal to $\mathbf{e}_1$ one thus ensures that this quasi-singular mode does not play a role anymore in the computation of $\mathbf{t}_k^{(D)}$.

This led us to consider the same example, but with a different initialization, for which $(\mathbf{r}_k)_1 = 0$ holds. More precisely, in this second situation, we take $\mathbf{u}_k$ equal to the Ritz vector associated to the smallest Ritz value from the subspace $\mathrm{span}\{\mathbf{e}_1, \mathbf{v}\}$, where $\mathbf{v}$ is defined as $\mathbf{u}_k$ in the first considered situation, that is,

$$\mathbf{v} = \sqrt{1 - s^2}\ \mathbf{x}_1 + s(I - \mathbf{x}_1\, \mathbf{x}_1^T)\, \mathbf{e}.$$

The results are reported in Figure 2. (Observe that here $\theta_k < a_{11}$ always holds.) Only little differences may now be seen between both methods, and the convergence rate is everywhere approximately equal to or less than α^2.

4. Theoretical analysis. We want to develop a mathematical analysis of the phenomena observed in the previous section. Note that in the considered experiment, $V_k = [\mathbf{u}_k]$; that is, θ_{k+1} is the smallest Ritz value associated to the subspace $\mathrm{span}\{\mathbf{u}_k, \mathbf{t}_k\}$. In general, Algorithm 2.1 extracts the approximate eigenpair from a larger subspace $\mathrm{span}(V_k) \supset \mathrm{span}\{\mathbf{u}_k, \mathbf{t}_k\}$. However, the smallest Ritz value associated to $\mathrm{span}(V_k)$ cannot be larger than that associated to $\mathrm{span}\{\mathbf{u}_k, \mathbf{t}_k\}$. An analysis of the latter thus gives an upper bound on the convergence rate for the general case.

Now, to develop this analysis, it is relevant to consider the projection $\widetilde{\mathbf{t}}_k^{(D)}$ of $\mathbf{t}_k^{(D)}$

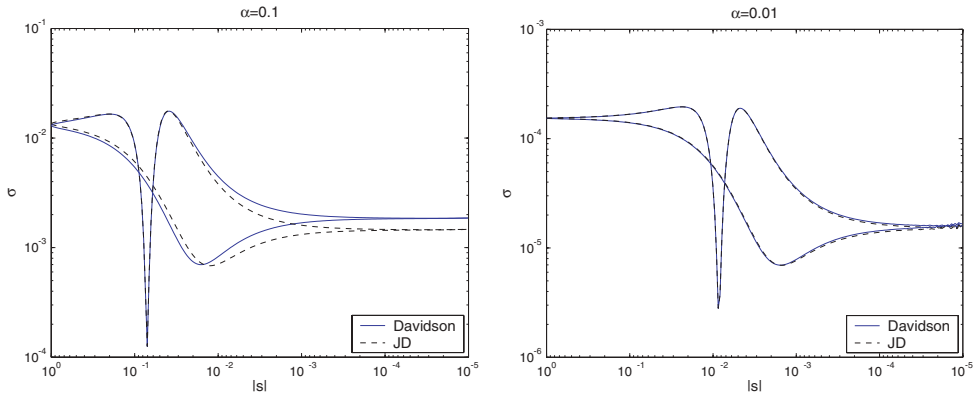


FIG. 2. *σ versus $|s|$ (second situation).*

onto $\mathbf{u}_k^\perp$:

$$\begin{aligned}\widetilde{\mathbf{t}}_k^{(D)} &= \mathbf{t}_k^{(D)} - \left(\mathbf{u}_k,\, \mathbf{t}_k^{(D)}\right)\, \mathbf{u}_k \\ &= M_k^{-1}\, \mathbf{r}_k - \left(\mathbf{u}_k,\, M_k^{-1}\, \mathbf{r}_k\right)\, \mathbf{u}_k. \end{aligned} \tag{4.1}$$

As mentioned at the beginning of section 2, a potential weakness of the Davidson method is that this vector converges to zero as M_k converges to $A - \theta_k\, I$. Besides the already mentioned argument that this cannot really happen if M_k is kept positive definite, we further observe that, as long as $\widetilde{\mathbf{t}}_k^{(D)}$ is not dominated by rounding errors, what matters is not its length but only its direction. We thus first need an upper bound on the smallest Ritz value from $\mathrm{span}\{\mathbf{u}_k,\, \mathbf{t}_k\}$ (for any given $\mathbf{t}_k \perp \mathbf{u}_k$), that would be more tractable than the exact expression, while being independent of the scaling of $\mathbf{t}_k$. This is the purpose of the following lemma.

LEMMA 4.1. *Let A be a real symmetric $n \times n$ matrix with eigenvalues $\lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$. Let $\mathbf{u}$ be a vector with unit norm such that the associated Rayleigh quotient*

$$\theta = (\mathbf{u},\, A\, \mathbf{u})$$

satisfies

$$\theta < \frac{\lambda_1 + \lambda_2}{2}. \tag{4.2}$$

For any vector $\mathbf{t} \perp \mathbf{u}$, the smallest Ritz value $\widetilde{\theta}$ associated to the subspace $\mathrm{span}\{\mathbf{u},\, \mathbf{t}\}$ *satisfies*

$$\frac{\widetilde{\theta} - \lambda_1}{\theta - \lambda_1} \leq 1 - \frac{(\mathbf{r},\, \mathbf{t})^2}{(\theta - \lambda_1)\, (\mathbf{t},\, (A - \theta\, I)\, \mathbf{t})\, \left(1 + \frac{\|\mathbf{r}\|^2}{(\lambda_1 + \lambda_2 - 2\,\theta)^2}\right)},$$

where $\mathbf{r} = A\, \mathbf{u} - \theta\, \mathbf{u}$.

Proof. Let $\hat{\theta}(\beta)$ be the Rayleigh quotient associated to a linear combination of the form

$$\hat{\mathbf{u}} \;=\; \frac{\mathbf{u} + \beta\, \mathbf{t}}{\sqrt{1 + \beta^2\, \|\mathbf{t}\|^2}}.$$

Since

$$\widetilde{\theta} \;=\; \min_{\beta} \hat{\theta}(\beta),$$

$\hat{\theta}\,(\beta)$ is an upper bound on $\widetilde{\theta}$ for any β. Now,

$$\hat{\theta}(\beta) - \theta \;=\; \frac{((\mathbf{u}+\beta\,\mathbf{t}),\,(A-\theta\,I)\,(\mathbf{u}+\beta\,\mathbf{t}))}{1+\beta^2\,\|\mathbf{t}\|^2}.$$

The numerator is minimal when

$$\beta \;=\; \frac{-\,((A-\theta\,I)\,\mathbf{u},\,\mathbf{t})}{(\mathbf{t},\,(A-\theta\,I)\,\mathbf{t})}, \tag{4.3}$$

and, since $(\mathbf{u},\,(A-\theta\,I)\,\mathbf{u}) = 0$, it is then equal to

$$\frac{-\,((A-\theta\,I)\,\mathbf{u},\,\mathbf{t})^2}{(\mathbf{t},\,(A-\theta\,I)\,\mathbf{t})} \;=\; \frac{-\,(\mathbf{r},\,\mathbf{t})^2}{(\mathbf{t},\,(A-\theta\,I)\,\mathbf{t})}.$$

Moreover, the denominator of the latter expression is positive when (4.2) holds because, by [13, Lemma 3.1],[1]

$$\min_{\substack{\mathbf{z}\,\perp\,\mathbf{u}\\ \|\mathbf{z}\|=1}} \;(\mathbf{z},\,(A-\theta\,I)\,\mathbf{z}) \;\geq\; \lambda_1+\lambda_2-2\,\theta, \tag{4.4}$$

showing that $(A-\theta\,I)$ is positive definite onto $\mathbf{u}^{\perp}$. On the other hand, for β given by (4.3), and using (4.4) again,

$$\begin{aligned}
\beta^2\,\|\mathbf{t}\|^2 &= \frac{((A-\theta\,I)\,\mathbf{u},\,\mathbf{t})^2}{(\mathbf{t},\,(A-\theta\,I)\,\mathbf{t})^2}\;\|\mathbf{t}\|^2\\
&\leq \|(A-\theta\,I)\,\mathbf{u}\|^2\;\frac{\|\mathbf{t}\|^4}{(\mathbf{t},\,(A-\theta\,I)\,\mathbf{t})^2}\\
&\leq \frac{\|\mathbf{r}\|^2}{(\lambda_1+\lambda_2-2\,\theta)^2}.
\end{aligned}$$

We thus find

$$\widetilde{\theta}-\theta \;\leq\; \frac{-\,(\mathbf{r},\,\mathbf{t})^2}{(\mathbf{t},\,(A-\theta\,I)\,\mathbf{t})\,\left(1+\frac{\|\mathbf{r}\|^2}{(\lambda_1+\lambda_2-2\,\theta)^2}\right)}, \tag{4.5}$$

whence the required result since $\frac{\widetilde{\theta}-\lambda_1}{\theta-\lambda_1} = 1+\frac{\widetilde{\theta}-\theta}{\theta-\lambda_1}$. ∎

We now develop the core of our quantitative analysis. First, in the following theorem, we prove a bound on the convergence rate considering the asymptotic situation of a matrix increasingly closer to a diagonal one. This is somewhat restrictive, but observe that this is precisely the situation for which we have little hint on how the method should behave when M_k is kept positive definite. The stated result is also in

[1] As pointed out by a referee, this inequality is also an obvious consequence of Cauchy's interlace theorem [18, p. 186], which implies that the sum of the p smallest eigenvalues of a symmetric matrix cannot be larger than the sum of all p Ritz values associated to any p dimensional subspace for any $1 \leq p \leq n$.

perfect agreement with the observations from Figure 1, although the matrix there is not that close to a diagonal one.

In this theorem, we use the assumption that all diagonal entries of the matrix are distinct. This is required because the proof makes use of standard perturbation analysis for matrices with distinct eigenvalues [29]; from a practical viewpoint, we suspect that only the separation of the smallest diagonal entry from the remaining entries really matters.

THEOREM 4.2. *Consider step k of Algorithm* 2.1 *applied to the real symmetric $n \times n$ matrix*

$$A \;=\; \mathrm{diag}(a_{ii}) \;+\; \varepsilon\, B,$$

where ε is a running parameter and B is a matrix such that $\mathrm{diag}(B) = 0$. *Assume that the diagonal entries of A are such that $a_{11} < a_{22} < \cdots < a_{nn}$ and that*

$$M_k \;=\; \mathrm{diag}(A) - \theta_k\, I$$

is positive definite. If

$$\mathbf{t}_k \;=\; M_k^{-1}\, \mathbf{r}_k,$$

then

$$\frac{\theta_{k+1} - \lambda_1}{\theta_k - \lambda_1} \;\leq\; \frac{1}{9} \;+\; \frac{8\sqrt{3}}{81}\,\delta \;+\; \frac{8\left(9 - \sqrt{3}\right)}{81}\,\delta^2 \;+\; \mathcal{O}(\varepsilon), \tag{4.6}$$

where

$$\delta \;=\; \sqrt{\frac{\theta_k - \lambda_1}{a_{11} - \lambda_1}}. \tag{4.7}$$

Proof. We first introduce some notation. We denote $(\mathbf{x}_i,\, \lambda_i)$ the eigenpairs of A (with $\lambda_1 \leq \cdots \leq \lambda_n$) and write

$$\mathbf{u}_k \;=\; c\,\mathbf{x}_1 \;+\; s\,\left(\sum_{j=2}^{n} \xi_j\,\mathbf{x}_j\right), \tag{4.8}$$

where $c^2 + s^2 = 1$ and $\sum_{j=2}^{n} \xi_j^2 = 1$; without loss of generality, we assume $c,\, s \geq 0$, the ξ_j being allowed to be positive or negative. The Rayleigh quotient and residual corresponding to $\mathbf{u}_k$ are, respectively,

$$\theta_k \;=\; \lambda_1 \;+\; s^2\,\left(\sum_{j=2}^{n} \xi_j^2 (\lambda_j - \lambda_1)\right) \tag{4.9}$$

and

$$\mathbf{r}_k \;=\; c\,(\lambda_1 - \theta_k)\,\mathbf{x}_1 \;+\; s\sum_{j=2}^{n} \xi_j\,(\lambda_j - \theta_k)\,\mathbf{x}_j. \tag{4.10}$$

We further define the following quantities:

$$\alpha = \sum_{j=2}^{n} \xi_j \, b_{1j}, \qquad \beta = \sum_{j=2}^{n} \frac{b_{1j}^2}{a_{jj} - a_{11}},$$
$$\gamma = \sum_{j=2}^{n} \xi_j^2 (a_{jj} - a_{11}), \qquad \eta = \frac{\alpha}{\sqrt{\beta\,\gamma}}.$$

We observe for future reference that, by virtue of the Cauchy–Schwarz inequality,

$$\eta^2 \;=\; \frac{\left(\sum_{j=2}^{n} \left(\frac{b_{1j}}{\sqrt{a_{jj}-a_{11}}}\right) \left(\xi_j \sqrt{a_{jj}-a_{11}}\right)\right)^2}{\left(\sum_{j=2}^{n} \frac{b_{1j}^2}{a_{jj}-a_{11}}\right) \left(\sum_{j=2}^{n} \xi_j^2 \, (a_{jj}-a_{11})\right)} \;\leq\; 1. \tag{4.11}$$

Now, the proof is rather technical but may be sketched as follows. We want to apply Lemma 4.1. As seen below, $\|\mathbf{r}_k\| = \mathcal{O}(\varepsilon)$. Hence we don't have to worry about the term $1 + \|\mathbf{r}\|^2/(\lambda_1 + \lambda_2 - 2\,\theta)^2$, and need only to bound below

$$\frac{\left(\mathbf{r}_k,\, \widetilde{\mathbf{t}}_k^{(D)}\right)^2}{(\theta - \lambda_1) \left(\widetilde{\mathbf{t}}_k^{(D)},\, (A - \theta\, I)\, \widetilde{\mathbf{t}}_k^{(D)}\right)}$$

with $\widetilde{\mathbf{t}}_k^{(D)} = M_k^{-1}\,\mathbf{r}_k - \left(\mathbf{u}_k,\, M_k^{-1}\,\mathbf{r}_k\right)\,\mathbf{u}_k$. This requires the knowledge of $M_k^{-1}\,\mathbf{r}_k$, which is a difficult task because $\mathbf{r}_k$ is naturally expressed in terms of the eigenvectors of A (see (4.10)), whereas the action of M_k^{-1} is easy to express for vectors defined in the standard canonical basis. However, because of the asymptotic situation considered, we may use perturbation theory to obtain an asymptotic expression for the eigenvectors $\mathbf{x}_i$ in term of the canonical vectors $\mathbf{e}_i$. From there, we obtain asymptotic expressions for $\mathbf{r}_k$, $M_k^{-1}\,\mathbf{r}_k$, $\mathbf{u}_k$, $\widetilde{\mathbf{t}}_k^{(D)}$, and finally for all the scalar quantities needed to apply Lemma 4.1. This gives an upper bound on the convergence rate that is valid up to $\mathcal{O}(\varepsilon)$. This upper bound depends on δ, on the matrix entries a_{jj} and b_{ij}, and on the unknown ξ_j. (It does not depend on s because the definition (4.7) of δ and (4.9) imply a relation between δ and s which we use to eliminate s.) Fortunately, the matrix entries and the ξ_j influence the result only through the variable η defined above; that is, the upper bound is a function of δ and η only. The required result is then proved by analyzing this function over the interval of interest, that is, $-1 \leq \eta \leq 1$ (see (4.11)) and

$$0 \;\leq\; \delta \;<\; 1,$$

as follows from (4.7) and from the positive definiteness of M_k, which implies $\theta_k \;<\; a_{11}$.

We now enter the core of the proof, stating first the needed results from perturbation theory and some immediate consequences. For $i = 1, \ldots, n$, [29, equations (9.4), (11.3), and (10.2)] yield

$$\lambda_i = a_{ii} \;+\; \varepsilon^2 \;\sum_{j \neq i} \frac{b_{ij}^2}{a_{ii} - a_{jj}} \;+\; \mathcal{O}(\varepsilon^3), \tag{4.12}$$

$$\mathbf{x}_i = \left(1 + \mathcal{O}(\varepsilon^2)\right) \mathbf{e}_i \;+\; \varepsilon \sum_{j \neq i} \left(\frac{b_{ij}}{a_{ii} - a_{jj}} + \mathcal{O}(\varepsilon)\right) \mathbf{e}_j. \tag{4.13}$$

Further, (4.12) implies

$$a_{11} - \lambda_1 = \varepsilon^2\,\beta\,(1+\mathcal{O}(\varepsilon)), \tag{4.14}$$

$$\lambda_i - \lambda_1 = (a_{ii} - a_{11})\,\left(1+\mathcal{O}(\varepsilon^2)\right), \quad i = 2,\ldots,n. \tag{4.15}$$

Hence (4.7) gives

$$\theta_k - \lambda_1 \;=\; \delta^2\,\varepsilon^2\,\beta\,(1+\mathcal{O}(\varepsilon)), \tag{4.16}$$

whereas (4.9) and (4.15) imply

$$\theta_k - \lambda_1 \;=\; s^2\,\left(\sum_{j=2}^{n} \xi_m^2 (a_{jj} - a_{11}) + \mathcal{O}(\varepsilon^2)\right) \;=\; s^2\,\gamma\,\left(1+\mathcal{O}(\varepsilon^2)\right).$$

Therefore, comparing with (4.16),

$$s \;=\; \delta\,\varepsilon\,\sqrt{\tfrac{\beta}{\gamma}}\,(1+\mathcal{O}(\varepsilon)), \tag{4.17}$$

which allows us to eliminate s in function of δ.

From these relations, we first deduce that, for ε sufficiently small, $a_{11} < a_{22}$ implies $\lambda_1 < \lambda_2$, whereas $\theta_k < a_{11}$ implies $\theta_k < (\lambda_1+\lambda_2)/2$. Thus, we may apply Lemma 4.1 whose assumptions are satisfied.

We now use the above relations to obtain asymptotic expressions for the needed vector quantities $\mathbf{r}_k$, $M_k^{-1}\,\mathbf{r}_k$, $\mathbf{u}_k$, and $\widetilde{\mathbf{t}}_k^{(D)}$. Observe that since the first diagonal entry of M_k is small ($\mathcal{O}(\varepsilon^2)$, see below), we need to correctly derive the leading term for first the component of $\mathbf{r}_k$, even though it is one order of magnitude smaller than the leading term of other components.

Noting that $c = 1+\mathcal{O}(s^2) = 1+\mathcal{O}(\varepsilon^2)$, we obtain, using first (4.16), (4.17), (4.15), and then (4.13)[2],

$$\begin{aligned}
\mathbf{r}_k &= -\delta^2\,\varepsilon^2\,\beta\,(1+\mathcal{O}(\varepsilon))\;\mathbf{x}_1 \;+\; \delta\,\varepsilon\,\sqrt{\tfrac{\beta}{\gamma}}\;\sum_{j=2}^{n} \xi_j\,(a_{jj}-a_{11})\;(1+\mathcal{O}(\varepsilon))\;\mathbf{x}_j \\
&= \left(-\delta^2\,\varepsilon^2\,\beta\,(1+\mathcal{O}(\varepsilon)) + \delta\,\varepsilon^2\,\sqrt{\tfrac{\beta}{\gamma}}\,\left(\sum_{j=2}^{n}\xi_j\,b_{j1} + \mathcal{O}(\varepsilon)\right)\right)\,\mathbf{e}_1 \\
&\qquad +\;\delta\,\varepsilon\,\sqrt{\tfrac{\beta}{\gamma}}\;\sum_{j=2}^{n}\left(\xi_j\,(a_{jj}-a_{11}) + \mathcal{O}(\varepsilon)\right)\,\mathbf{e}_j \\
&= \delta\,\varepsilon^2\,\beta\,(\eta - \delta + \mathcal{O}(\varepsilon))\,\mathbf{e}_1 \;+\; \delta\,\varepsilon\,\sqrt{\tfrac{\beta}{\gamma}}\;\sum_{j=2}^{n}\left(\xi_j\,(a_{jj}-a_{11}) + \mathcal{O}(\varepsilon)\right)\,\mathbf{e}_j.
\end{aligned} \tag{4.18}$$

Incidentally, this also shows that $\|\mathbf{r}_k\| = \mathcal{O}(\varepsilon)$, as claimed above.

Since $\delta^2(a_{11}-\theta_k) = (1-\delta^2)(\theta_k - \lambda_1)$ (from (4.7)), one has, with (4.16),

$$\theta_k = a_{11} - (1-\delta^2)\varepsilon^2\,\beta\,(1+\mathcal{O}(\varepsilon)).$$

[2]In which we may consider only the zero order terms and the first order term of $\mathbf{x}_j$ ($j \neq 1$) along $\mathbf{e}_1$, all other first order terms leading to $\mathcal{O}(\varepsilon^2)$ contributions to the coefficients of $\mathbf{e}_k$ for $k \neq 1$.

Therefore,

$$(M_k)_{jj} \;=\; a_{jj} - \theta_k \;=\; \begin{cases} (1-\delta^2)\varepsilon^2\,\beta\,(1+\mathcal{O}(\varepsilon)) & \text{if } j=1, \\ (a_{jj}-a_{11})\,\left(1+\mathcal{O}(\varepsilon^2)\right) & \text{otherwise,} \end{cases}$$

and we further obtain

$$M_k^{-1}\,\mathbf{r}_k \;=\; \frac{\delta\,(\eta-\delta+\mathcal{O}(\varepsilon))}{1-\delta^2}\,\mathbf{e}_1 \;+\; \delta\,\varepsilon\,\sqrt{\tfrac{\beta}{\gamma}}\,\sum_{j=2}^{n}(\xi_j+\mathcal{O}(\varepsilon))\,\mathbf{e}_j. \tag{4.19}$$

On the other hand, we find, with (4.8), (4.17), and (4.13),

$$\mathbf{u}_k \;=\; \left(1+\mathcal{O}(\varepsilon^2)\right)\,\mathbf{e}_1 \;+\; \varepsilon\,\left(\sum_{j=2}^{n}\left(\delta\,\sqrt{\tfrac{\beta}{\gamma}}\,\xi_j - \tfrac{b_{1j}}{a_{jj}-a_{11}} + \mathcal{O}(\varepsilon)\right)\,\mathbf{e}_j\right),$$

from which we deduce

$$\left(\mathbf{u}_k,\, M_k^{-1}\,\mathbf{r}_k\right) \;=\; \left(\mathbf{e}_1,\, M_k^{-1}\,\mathbf{r}_k\right) \;+\; \mathcal{O}(\varepsilon^2)$$

and, therefore,

$$\left(\mathbf{e}_1,\, \widetilde{\mathbf{t}}_k^{(D)}\right) \;=\; \left(\mathbf{e}_1,\, M_k^{-1}\,\mathbf{r}_k\right)\,(1-(\mathbf{e}_1,\,\mathbf{u}_k)) + \mathcal{O}(\varepsilon^2)(\mathbf{e}_1,\,\mathbf{u}_k) \;=\; \mathcal{O}(\varepsilon^2).$$

Hence,

$$\widetilde{\mathbf{t}}_k^{(D)} \;=\; \mathcal{O}(\varepsilon^2)\,\mathbf{e}_1 \;+\; \varepsilon\,\delta\,\left(\sum_{j=2}^{n}\left(\sqrt{\tfrac{\beta}{\gamma}}\,\xi_j\,\tfrac{1-\delta\,\eta}{1-\delta^2} + \tfrac{\eta-\delta}{1-\delta^2}\,\tfrac{b_{1j}}{a_{jj}-a_{11}} + \mathcal{O}(\varepsilon)\right)\,\mathbf{e}_j\right).$$

We are now able to derive asymptotic expressions for the inner products needed to apply Lemma 4.1. We find, remembering the definition of α, β, γ, η,

$$\begin{aligned}\left(\mathbf{r}_k,\, \widetilde{\mathbf{t}}_k^{(D)}\right) &= \varepsilon^2\,\delta^2\,\sqrt{\tfrac{\beta}{\gamma}}\,\left(\sqrt{\tfrac{\beta}{\gamma}}\,\tfrac{1-\delta\,\eta}{1-\delta^2}\,\gamma \;+\; \tfrac{\eta-\delta}{1-\delta^2}\,\alpha \;+\; \mathcal{O}(\varepsilon)\right) \\ &= \varepsilon^2\,\delta^2\,\beta\,(\zeta_1+\mathcal{O}(\varepsilon)),\end{aligned}$$

where $\zeta_1 = \frac{1+\eta^2-2\,\eta\,\delta}{1-\delta^2}$. Moreover,

$$\begin{aligned}&\left(\widetilde{\mathbf{t}}_k^{(D)},\, (A-\theta_k\,I)\,\widetilde{\mathbf{t}}_k^{(D)}\right) \\ &\quad= \left(\widetilde{\mathbf{t}}_k^{(D)},\, (\mathrm{diag}(A)-a_{11}\,I)\,\widetilde{\mathbf{t}}_k^{(D)} + \mathcal{O}(\varepsilon)\right) \\ &\quad= \varepsilon^2\,\delta^2\,\left(\beta\,\left(\tfrac{1-\delta\,\eta}{1-\delta^2}\right)^2 \;+\; \left(\tfrac{\eta-\delta}{1-\delta^2}\right)^2\,\beta \;+\; 2\,\tfrac{1-\delta\,\eta}{1-\delta^2}\,\tfrac{\eta-\delta}{1-\delta^2}\,\sqrt{\tfrac{\beta}{\gamma}}\,\alpha + \mathcal{O}(\varepsilon)\right) \\ &\quad= \varepsilon^2\,\delta^2\,\beta\,\left(\frac{(1-\delta\,\eta)^2+(\eta-\delta)^2+2\,\eta\,(1-\delta\,\eta)(\eta-\delta)}{(1-\delta^2)^2} + \mathcal{O}(\varepsilon)\right) \\ &\quad= \varepsilon^2\,\delta^2\,\beta\,\left(\zeta_2+\zeta_1^2+\mathcal{O}(\varepsilon)\right), \qquad (4.20)\end{aligned}$$

where $\zeta_2 = \frac{(\eta-\delta)^2\,(1-\eta^2)}{1-\delta^2}$. Therefore, Lemma 4.1 gives, using these expressions and (4.16), and remembering also that $\|\mathbf{r}_k\| = \mathcal{O}(\varepsilon)$ (see (4.18)),

$$\begin{aligned}\frac{\theta_{k+1}-\lambda_1}{\theta_k-\lambda_1} &\le 1 - \frac{(\zeta_1+\mathcal{O}(\varepsilon))^2}{(\zeta_1^2+\zeta_2+\mathcal{O}(\varepsilon))(1+\mathcal{O}(\varepsilon))} \\ &= 1 - \frac{\zeta_1^2}{\zeta_1^2+\zeta_2}\left(1+\zeta_1^{-1}\mathcal{O}(\varepsilon)\right)\left(1+(\zeta_1^2+\zeta_2)^{-1}\mathcal{O}(\varepsilon)\right)\left(1+\mathcal{O}(\varepsilon)\right).\end{aligned}$$

Since $\zeta_2 \geq 0$ and $\zeta_1 = 1 + (\eta - \delta)^2/(1 - \delta^2) \geq 1$, the $\mathcal{O}(\varepsilon)$ terms in the right-hand side are harmless because they simply correspond to the $\mathcal{O}(\varepsilon)$ error term in the result (4.6) to be proved. Letting $g(\delta) = \frac{1}{9} + \frac{8\sqrt{3}}{81}\,\delta + \frac{8(9-\sqrt{3})}{81}\,\delta^2$, we are thus left with proving that

$$\left(1 + \frac{\zeta_1}{\zeta_2}\right)^{-1} \leq g(\delta),$$

i.e., that

$$f(\delta,\, \eta) = \frac{\zeta_2}{\zeta_1}\;\frac{1 - g(\delta)}{g(\delta)} = \frac{(\eta - \delta)^2\,(1 - \eta^2)}{(1 + \eta^2 - 2\,\eta\,\delta)^2}\;\frac{1 - g(\delta)}{g(\delta)}$$

does not exceed 1 for $0 \leq \delta < 1$ and $-1 \leq \eta \leq 1$. This function is continuous and infinitely derivable over this interval. A fine sampling and surface plot further reveals that it is smooth, has two maxima located around $\delta = 0$, $\eta = \pm 0.57$, and that it is certainly less than 1 outside the neighborhood of these maxima. Consider then the function $f(0,\, \eta) = 8(1 - \eta^2)\,\eta^2/(1 + \eta^2)^2$. As easily checked, it is maximal for $\eta^2 = 1/3$ and $f(0,\, \pm\frac{1}{\sqrt{3}}) = 1$. Thus, the required result holds if $(\delta,\, \eta) = (0,\, \frac{1}{\sqrt{3}})$ and $(\delta,\, \eta) = (0,\, -\frac{1}{\sqrt{3}})$ correspond to maxima of $f(\delta,\, \eta)$ over the region of interest. To check this, we used computer algebra to obtain the Taylor expansion around these points. This gives

$$f = 1 - 2\,\sqrt{3}\,\delta - \tfrac{27}{2}\,\left(\eta - \tfrac{1}{\sqrt{3}}\right)^2 \;+\; \mathcal{O}\left(\delta^2 + \delta\,|\eta - \tfrac{1}{\sqrt{3}}| + \left(\eta - \tfrac{1}{\sqrt{3}}\right)^2\right)$$

for the first point and

$$f = 1 + \frac{1}{2}\,\left(\delta\eta + \tfrac{1}{\sqrt{3}}\right)\begin{pmatrix} \frac{-121 + 12\,\sqrt{3}}{6} & \frac{15}{2} \\ \frac{15}{2} & -\frac{27}{2} \end{pmatrix}\begin{pmatrix} \delta \\ \eta + \frac{1}{\sqrt{3}} \end{pmatrix} \;+\; \mathcal{O}\left(\left(\delta + |\eta + \tfrac{1}{\sqrt{3}}|\right)^2\right)$$

for the second one, showing that both points correspond indeed to maxima of $f(\eta,\, \delta)$ over the region $-1 \leq \eta \leq 1$, $0 \leq \delta < 1$, which concludes the proof. □

This theorem essentially proves that, if a matrix is sufficiently close to a diagonal one, then the Davidson method converges with a convergence rate bounded away from 1 as soon as δ is away from 1 (that is, θ_k away from a_{11}). Further, the asymptotic convergence rate for $\delta \to 0$ (i.e., $\theta_k \to \lambda_1$) is not larger than $\frac{1}{9}$. Remarkably, this bound does not depend on the matrix entries. This is in perfect agreement with the observations from Figure 1, where the asymptotic convergence rate (for $\theta_k \to \lambda_1$) was found to be slightly less than $\frac{1}{10}$ and essentially independent of the size of the offdiagonal entries.

In the theorem, we assume that all computations are done exactly. One could then wonder how small ε can be before the results get obscured by rounding errors. Here, the most dangerous step is certainly the orthogonalization of $\mathbf{t}_k^{(D)}$ against the previous vectors at substep 8 of Algorithm 2.1, which results in dramatic cancellation when $\mathbf{t}_k^{(D)}$ is strongly aligned with $\mathbf{u}_k$. One has, therefore, to expect severe roundoff effects when

$$\|\widetilde{\mathbf{t}}_k^{(D)}\| \;\ll\; \|\mathbf{t}_k^{(D)}\| \;=\; \|M_k^{-1}\,\mathbf{r}_k\|,$$

where $\widetilde{\mathbf{t}}_k^{(D)}$ is the orthogonal projection of $\widetilde{\mathbf{t}}_k^{(D)}$ against $\mathbf{u}_k^\perp$ (see (4.1)). Interestingly, the quantities in this relation may be estimated from the proof of Theorem 4.2. Neglecting $\mathcal{O}(\varepsilon)$ terms, this gives (see Appendix A for the details of the calculation)

$$\frac{\|\widetilde{\mathbf{t}}_k^{(D)}\|}{\|M_k^{-1}\,\mathbf{r}_k\|} \;\geq\; \frac{\varepsilon\,(1-\delta)^2}{2}\;\left(\sum_{j=2}^{n}\frac{b_{1j}^2}{(a_{jj}-a_{11})(a_{nn}-a_{11})}\right)^{1/2}. \tag{4.21}$$

Thus, when δ is away from 1 (i.e., θ_k away from a_{11}), severe roundoff effects are not expected before the offdiagonal entries in A become very small. (To figure how small they can be, see a related discussion in [25, section 4.3]). This is somewhat surprising, but remember that we assume M_k is positive definite, and the picture might be well different for indefinite M_k.

Now, these results do not explain the observations from Figure 2. For this purpose, we need a further analysis that exploits the relation $(\mathbf{r}_k)_1 = 0$ to obtain a better bound, proving increasing convergence speed as the matrix becomes closer to a diagonal one. This is done in Theorem 4.3 below. Observe that the result is here not restricted to the asymptotic situation of a matrix converging to a diagonal one.

THEOREM 4.3. *Consider step k of Algorithm 2.1 applied to a real symmetric $n\times n$ matrix A whose eigenpairs $(\lambda_i,\,\mathbf{x}_i)$ are such that $\lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$ and whose diagonal entries satisfy $a_{11} < a_{22} \leq \cdots \leq a_{nn}$. Assume that*

$$M_k \;=\; \mathrm{diag}(A) - \theta_k\, I$$

is positive definite and that

$$\theta_k \;<\; \frac{\lambda_1+\lambda_2}{2}. \tag{4.22}$$

Let

$$\delta \;=\; \sqrt{\frac{\theta_k-\lambda_1}{a_{11}-\lambda_1}}$$

and

$$N = \mathrm{offdiag}(A).$$

If

$$\mathbf{t}_k \;=\; M_k^{-1}\,\mathbf{r}_k,$$

$$(\mathbf{r}_k)_1 \;=\; 0, \tag{4.23}$$

and

$$\frac{\delta}{\sqrt{1-\delta^2}} \;\leq\; \frac{1}{\sqrt{1+\frac{\|N\|}{\lambda_2-\lambda_1}}}, \tag{4.24}$$

then

$$\frac{\theta_{k+1}-\lambda_1}{\theta_k-\lambda_1} \;\leq\; 1 - \frac{\left(1-\frac{\delta}{\sqrt{1-\delta^2}}\,\sqrt{1+\frac{\|N\|}{\lambda_2-\lambda_1}}\right)^2}{\left(1+\frac{\|\mathbf{r}_k\|^2}{(\lambda_1+\lambda_2-2\,\theta_k)^2}\right)\,\left(1+\frac{\|\mathbf{r}_k\|+2\,\|N\|}{a_{22}-a_{11}}\right)\left(1+\frac{\|N\|}{\lambda_2-\lambda_1}\right)} \tag{4.25}$$

and, for $\|\mathbf{r}_k\| \to 0$,

$$\frac{\theta_{k+1}-\lambda_1}{\theta_k-\lambda_1} \;\le\; \|N\| \left(\frac{1}{\lambda_2-\lambda_1}+\frac{2}{a_{22}-a_{11}}\right) \left(1+\mathcal{O}\left(\|\mathbf{r}_k\|\right)\right). \tag{4.26}$$

Proof. All assumptions of Lemma 4.1 are satisfied, and its application yields (4.25) if we are able to prove that

$$\frac{\left(\mathbf{r}_k,\, \widetilde{\mathbf{t}}_k^{(D)}\right)^2}{(\theta-\lambda_1)\left(\widetilde{\mathbf{t}}_k^{(D)},\, (A-\theta_k\, I)\,\widetilde{\mathbf{t}}_k^{(D)}\right)} \;\ge\; \frac{\left(\frac{1}{\sqrt{1+\frac{\|N\|}{\lambda_2-\lambda_1}}}-\frac{\delta}{\sqrt{1-\delta^2}}\right)^2}{1+\frac{\|\mathbf{r}_k\|+2\,\|N\|}{a_{22}-a_{11}}}, \tag{4.27}$$

where $\widetilde{\mathbf{t}}_k^{(D)} \;=\; M_k^{-1}\,\mathbf{r}_k - \left(\mathbf{u}_k,\, M_k^{-1}\,\mathbf{r}_k\right)\,\mathbf{u}_k$. Here, note that $(\mathbf{r}_k,\, \mathbf{u}_k)=0$ implies

$$\left(\mathbf{r}_k,\, \widetilde{\mathbf{t}}_k^{(D)}\right) \;=\; (\mathbf{r}_k,\, M_k^{-1}\,\mathbf{r}_k). \tag{4.28}$$

The proof may then be sketched as follows. First, $\left(\widetilde{\mathbf{t}}_k^{(D)},\, (A-\theta_k\, I)\,\widetilde{\mathbf{t}}_k^{(D)}\right)$ is bounded above in function of $(\mathbf{r}_k,\, M_k^{-1}\,\mathbf{r}_k)$. Considering (4.27), we are then left with the analysis of $(\mathbf{r}_k,\, M_k^{-1}\,\mathbf{r}_k)/(\theta_k-\lambda_1)$, which is developed next.

One has

$$\begin{aligned}(A-\theta_k\, I)\,\widetilde{\mathbf{t}}_k^{(D)} &= (M_k+N)\,M_k^{-1}\,\mathbf{r}_k \;-\; (\mathbf{u}_k,\, M_k^{-1}\,\mathbf{r}_k)\,\mathbf{r}_k \\ &= \left(1-(\mathbf{u}_k,\, M_k^{-1}\,\mathbf{r}_k)\right)\mathbf{r}_k \;+\; N\,M_k^{-1}\,\mathbf{r}_k,\end{aligned}$$

whence

$$\begin{aligned}\left(\widetilde{\mathbf{t}}_k^{(D)},\, (A-\theta_k\, I)\,\widetilde{\mathbf{t}}_k^{(D)}\right) &= (\mathbf{r}_k,\, M_k^{-1}\,\mathbf{r}_k)\left(1-(\mathbf{u}_k,\, M_k^{-1}\,\mathbf{r}_k)\right) \\ &\quad + (M_k^{-1}\,\mathbf{r}_k,\, N\,M_k^{-1}\,\mathbf{r}_k) \;-\; (\mathbf{u}_k,\, M_k^{-1}\,\mathbf{r}_k)\,(\mathbf{u}_k,\, N\,M_k^{-1}\,\mathbf{r}_k).\end{aligned}$$

Further, since N is symmetric,

$$\begin{aligned}|(\mathbf{u}_k,\, M_k^{-1}\,\mathbf{r}_k)| &\le \|M_k^{-1}\,\mathbf{r}_k\|, \\ |(M_k^{-1}\,\mathbf{r}_k,\, N\,M_k^{-1}\,\mathbf{r}_k)| &\le \|N\|\;\|M_k^{-1}\,\mathbf{r}_k\|^2, \\ |(\mathbf{u}_k,\, N\,M_k^{-1}\,\mathbf{r}_k)| &\le \|N\,M_k^{-1}\,\mathbf{r}_k\| \\ &\le \|N\|\;\|M_k^{-1}\,\mathbf{r}_k\|.\end{aligned}$$

Therefore, since by virtue of (4.23)

$$\|M_k^{-1}\,\mathbf{r}_k\| \;\le\; \sqrt{\frac{(\mathbf{r}_k,\, M_k^{-1}\,\mathbf{r}_k)}{a_{22}-a_{11}}} \;\le\; \frac{\|\mathbf{r}_k\|}{a_{22}-a_{11}},$$

we obtain

$$\left(\widetilde{\mathbf{t}}_k^{(D)},\, (A-\theta_k\, I)\,\widetilde{\mathbf{t}}_k^{(D)}\right) \;\le\; (\mathbf{r}_k,\, M_k^{-1}\,\mathbf{r}_k)\;\left(1+\frac{\|\mathbf{r}_k\|+2\,\|N\|}{a_{22}-a_{11}}\right).$$

With (4.28), this gives

$$\frac{1}{\theta_k-\lambda_1}\;\frac{\left((A-\theta_k\, I)\,\mathbf{u}_k,\, \widetilde{\mathbf{t}}_k^{(D)}\right)^2}{\left(\widetilde{\mathbf{t}}_k^{(D)},\, (A-\theta_k\, I)\,\widetilde{\mathbf{t}}_k^{(D)}\right)} \;\ge\; \frac{(\mathbf{r}_k,\, M_k^{-1}\,\mathbf{r}_k)}{(\theta_k-\lambda_1)\left(1+\frac{\|\mathbf{r}_k\|+2\,\|N\|}{a_{22}-a_{11}}\right)}. \tag{4.29}$$

Comparing with (4.27), we are thus, as stated above, left with the analysis of $(\mathbf{r}_k,\, M_k^{-1}\,\mathbf{r}_k)/(\theta_k-\lambda_1)$. For this purpose, let $\mathbf{u}_\perp = (I-\mathbf{x}_1\,\mathbf{x}_1^T)\mathbf{u}_k$. Since $(\mathbf{x}_1,\,\mathbf{u}_\perp)=0$ and $\mathbf{u}_k = c\,\mathbf{x}_1+\mathbf{u}_\perp$ (with $c=(\mathbf{x}_1,\,\mathbf{u}_k)$), one has

$$\mathbf{r}_k \;=\; (A-\lambda_1)\,\mathbf{u}_\perp \;-\; (\theta_k-\lambda_1)\,\mathbf{u}_k,$$

whence

$$\frac{\|\mathbf{r}_k\|_{M_k^{-1}}}{\sqrt{\theta_k-\lambda_1}} \;\geq\; \frac{\|(A-\lambda_1)\,\mathbf{u}_\perp\|_{M_k^{-1}}}{\sqrt{\theta_k-\lambda_1}} \;-\; \sqrt{\theta_k-\lambda_1}\,\|\mathbf{u}_k\|_{M_k^{-1}}.$$

Therefore, since

$$\theta_k-\lambda_1 \;=\; (\mathbf{u}_\perp,\,(A-\lambda_1)\,\mathbf{u}_\perp),$$

and $\|M_k^{-1}\| \leq (a_{11}-\theta_k)^{-1}$, one finds

$$\begin{aligned}\frac{\|\mathbf{r}_k\|_{M_k^{-1}}}{\sqrt{\theta_k-\lambda_1}} &\geq \frac{(\mathbf{u}_\perp,\,(A-\lambda_1)\,M_k^{-1}\,(A-\lambda_1)\,\mathbf{u}_\perp)^{1/2}}{(\mathbf{u}_\perp,\,(A-\lambda_1)\,\mathbf{u}_\perp)^{1/2}} \;-\; \sqrt{\frac{\theta_k-\lambda_1}{a_{11}-\theta_k}} \\ &\geq \sqrt{\nu_{\min}} \;-\; \frac{\delta}{\sqrt{1-\delta^2}}, \qquad (4.30)\end{aligned}$$

where $\nu_{\min}$ stands for the smallest nonzero eigenvalue of the pencil $(A-\lambda_1\,I)-\nu\,M_k$, i.e.,

$$\begin{aligned}\nu_{\min} &= \min_{\substack{\mathbf{z}\,\perp\,\mathbf{x}_1\\ \mathbf{z}\neq 0}} \frac{(\mathbf{z},\,(A-\lambda_1\,I)\,\mathbf{z})}{(\mathbf{z},\,M_k\,\mathbf{z})} \\ &\geq \min_{\substack{\mathbf{z}\,\perp\,\mathbf{x}_1\\ \mathbf{z}\neq 0}} \frac{(\mathbf{z},\,(A-\lambda_1\,I)\,\mathbf{z})}{(\mathbf{z},\,(\mathrm{diag}(A)-\lambda_1\,I)\,\mathbf{z})} \\ &= \min_{\substack{\mathbf{z}\,\perp\,\mathbf{x}_1\\ \mathbf{z}\neq 0}} \left(1-\frac{(\mathbf{z},\,N\,\mathbf{z})}{(\mathbf{z},\,(A-\lambda_1\,I)\,\mathbf{z})}\right)^{-1} \\ &\geq \left(1+\frac{\|N\|}{\lambda_2-\lambda_1}\right)^{-1}.\end{aligned}$$

With (4.29) and (4.30), this proves (4.27) and, therefore, (4.25).

On the other hand, by [19, Lemma 3.2], $(\theta_k-\lambda_1)(\lambda_2-\theta_k) \leq \|\mathbf{r}_k\|^2$. Hence, for $\|\mathbf{r}_k\| \to 0$, $\delta = \mathcal{O}(\|\mathbf{r}_k\|)$ and (4.25) yields

$$\frac{\theta_{k+1}-\lambda_1}{\theta_k-\lambda_1} \;\leq\; 1-\frac{1+\mathcal{O}\left(\|\mathbf{r}_k\|\right)}{\left(1+\frac{2\,\|N\|}{a_{22}-a_{11}}\right)\left(1+\frac{\|N\|}{\lambda_2-\lambda_1}\right)}.$$

Inequality (4.25) then readily follows because $1-1/((1+x)(1+y)) \leq x+y$ for any nonnegative x, y. ∎

Theorem 4.3 explains, at least qualitatively, the observations from Figure 2: When $(\mathbf{r}_k)_1 = 0$ holds, the convergence speed is no longer independent of the matrix entries but increases as the matrix becomes closer to a diagonal one. However, our result seems too pessimistic, at least with respect to the situation depicted in Figure 2, which suggests that the convergence rate is then $\mathcal{O}\left(\alpha^2\right) = \mathcal{O}\left(\|N\|^2\right)$, whereas (4.26) proves only $\mathcal{O}\left(\|N\|\right)$ convergence.

Now, it is worth comparing Theorems 4.2 and 4.3 with the known results applicable to the JD method [14, 16, 21, 27]. Among these, the easiest to express and to interpret is the one obtained by combining [14, Theorem 3.1] with [14, equation (4.3)]. Assuming only

$$\theta_k \;<\; \frac{\lambda_1+\lambda_2}{2}, \tag{4.31}$$

this indeed yields

$$\frac{\theta_{k+1}-\lambda_1}{\lambda_2-\theta_{k+1}} \;\le\; \left(\frac{(\theta_k-\lambda_1)+\gamma\,(\lambda_2-\theta_k)}{(\lambda_2-\theta_k)+\gamma\,(\theta_k-\lambda_1)}\right)^2 \frac{\theta_k-\lambda_1}{\lambda_2-\theta_k}, \tag{4.32}$$

where γ is the relative error left in the correction equation (2.3), measured with respect to the energy norm. (Observe that by (4.4) the system matrix of the correction equation is indeed positive definite onto $\mathbf{u}_k^\perp$ when (4.31) holds.) Since the scaling of $\mathbf{t}_k$ is unimportant anyway, the case (2.6) where one computes $\mathbf{t}_k^{(JD)}$ with just one application of the preconditioner is actually equivalent to the situation where one performs a single steepest descent iteration. Hence (see, e.g., [1, Theorem 1.8]),

$$\gamma \;\le\; \frac{\kappa-1}{\kappa+1}, \tag{4.33}$$

where

$$\kappa \;=\; \frac{\max_{\substack{\mathbf{z}\,\perp\,\mathbf{u}_k\\ \mathbf{z}\neq 0}} \frac{(\mathbf{z},\,(A-\theta_k\,I)\,\mathbf{z})}{(\mathbf{z},\,M_k\,\mathbf{z})}}{\min_{\substack{\mathbf{z}\,\perp\,\mathbf{u}_k\\ \mathbf{z}\neq 0}} \frac{(\mathbf{z},\,(A-\theta_k\,I)\,\mathbf{z})}{(\mathbf{z},\,M_k\,\mathbf{z})}}$$

is the condition number of the system (2.3) preconditioned by (2.4). Further, letting

$$\widetilde{N} = A-\theta_k\,I-M_k$$

(i.e., $\widetilde{N}=N$ is the offdiagonal part of A whenever using the diagonal preconditioning (2.2)), one has, using (4.4),

$$\begin{aligned}
\kappa &= \frac{\max_{\substack{\mathbf{z}\,\perp\,\mathbf{u}_k\\ \mathbf{z}\neq 0}} \left(1 \;-\; \frac{(\mathbf{z},\,\widetilde{N}\,\mathbf{z})}{(\mathbf{z},\,(A-\theta_k\,I)\,\mathbf{z})}\right)^{-1}}{\min_{\substack{\mathbf{z}\,\perp\,\mathbf{u}_k\\ \mathbf{z}\neq 0}} \left(1 \;+\; \frac{(\mathbf{z},\,\widetilde{N}\,\mathbf{z})}{(\mathbf{z},\,(A-\theta_k\,I)\,\mathbf{z})}\right)^{-1}} \\
&\le \frac{\max_{\substack{\mathbf{z}\,\perp\,\mathbf{u}_k\\ \mathbf{z}\neq 0}} \left(1 \;-\; \|\widetilde{N}\|\,\frac{(\mathbf{z},\,\mathbf{z})}{(\mathbf{z},\,(A-\theta_k\,I)\,\mathbf{z})}\right)^{-1}}{\min_{\substack{\mathbf{z}\,\perp\,\mathbf{u}_k\\ \mathbf{z}\neq 0}} \left(1 \;+\; \|\widetilde{N}\|\,\frac{(\mathbf{z},\,\mathbf{z})}{(\mathbf{z},\,(A-\theta_k\,I)\,\mathbf{z})}\right)^{-1}} \\
&\le \frac{1 \;+\; \frac{\|\widetilde{N}\|}{\lambda_1+\lambda_2-2\,\theta_k}}{1 \;-\; \frac{\|\widetilde{N}\|}{\lambda_1+\lambda_2-2\,\theta_k}}.
\end{aligned}$$

This yields, with (4.33),

$$\gamma \;\le\; \frac{\|\widetilde{N}\|}{\lambda_1+\lambda_2-2\,\theta_k},$$

which may be combined with (4.32) to obtain an upper bound on the convergence rate comparable to the ones in Theorems 4.2 and 4.3. In particular, for $\theta_k \to \lambda_1$, this gives

$$\frac{\theta_{k+1} - \lambda_1}{\theta_k - \lambda_1} \;\leq\; \frac{\|\widetilde{N}\|^2}{\lambda_2 - \lambda_1} \left(1 + \mathcal{O}(\theta_k - \lambda_1)\right).$$

These results indicate that the JD method may improve the Davidson method in several ways. First, the above analysis of the JD method holds independently of the positive definiteness of M_k; only (4.31) has to be assumed, which is much weaker for nearly diagonal matrices. Next, a convergence estimate that vanishes for $\theta_k \to \lambda_1$ and $\|\widetilde{N}\| \to 0$ is obtained without having to enforce any particular condition on the residual such as $(\mathbf{r}_k)_1 = 0$. Further, the latter estimate is $\mathcal{O}(\|\widetilde{N}\|^2)$, whereas (4.25) proves only $\mathcal{O}\left(\|N\|\right)$ convergence for the Davidson method. Finally, the above analysis of the JD method applies to any M_k and corresponding $\widetilde{N}$, whereas Theorem 4.3 is restricted to diagonal preconditioning. (The latter restriction is actually connected to the assumption $(\mathbf{r}_k)_1 = 0$. We could indeed rewrite Theorem 4.2 so that it applies to general preconditioning of the form $M_k = G - \theta_k I$, but then $(\mathbf{r}_k)_1 = 0$ should be rewritten $(\mathbf{w}_1, \mathbf{r}_k) = 0$, where $\mathbf{w}_1$ is the first eigenvector of G. To figure this out, consider a basis transformation that makes G diagonal, apply the theorem in its present form to the tranformed matrix, and bring back the result in the original basis. We did not consider this as it would be of little practical interest.)

Now, in Figure 2, we see only little difference between the JD and the Davidson methods initialized in the standard way. We explain this as follows. First, $\theta_k < a_{11}$ and $(\mathbf{r}_k)_1 = 0$ then always hold because $\mathbf{e}_1$ belongs to the starting subspace. Further, the difference between the $\mathcal{O}(\|\widetilde{N}\|^2)$ convergence estimate for the JD method and the $\mathcal{O}\left(\|N\|\right)$ one for the Davidson method might well come from a shortcoming in our analysis. Indeed, $(\mathbf{r}_k)_1 = 0$ entails $(M_k\,\mathbf{r}_k)_1 = 0$ (remember that M_k is diagonal), whereas $\mathbf{x}_1 \approx \mathbf{e}_1$ for nearly diagonal matrices with $\mathbf{u}_k \approx \mathbf{x}_1$ when one is close to convergence. Hence, $(\mathbf{u}_k,\, M_k^{-1}\,\mathbf{r}_k) \approx 0$, entailing $\mathbf{t}_k^{(D)} \approx -\mathbf{t}_k^{(JD)}$ (compare (2.1) and (2.6)). One even has $\mathbf{t}_k^{(D)} = -\mathbf{t}_k^{(JD)}$ when the basis of $\mathrm{span}(V_k)$ contains only conical vectors (as occurs at the first step of Algorithm 2.1 with the standard initialization strategy). Indeed, then $\mathbf{r}_k \in \mathrm{span}(V_k)^\perp$ entails $M_k^{-1}\,\mathbf{r}_k \in \mathrm{span}(V_k)^\perp$, whence $(\mathbf{u}_k,\, M_k^{-1}\,\mathbf{r}_k) = 0$. In general, when $(\mathbf{r}_k)_1 = 0$, the Davidson method corresponds anyway to a (nonstandard) JD scheme, with skew projection $\left(I - \mathbf{u}_k(\mathbf{u}_k,\, \mathbf{e}_1)^{-1}\mathbf{e}_1^T\right)$ instead of the orthogonal projection $\left(I - \mathbf{u}_k\,\mathbf{u}_k^T\right)$; see [22, section 7.1.1] for details and discussion.

Therefore, the JD approach is mainly helpful in "nonstandard" situations. For instance, it is more robust with respect to the choice of the initial approximation. It may also be better whenever using nondiagonal preconditioners of the form (2.7). Indeed, nice guaranteed convergence of the Davidson method requires the positive definiteness of M_k, which, as mentioned at the end of section 2, is more difficult to ensure from the beginning with general (nondiagonal) preconditioning. Moreover, the initialization strategy that allows the Davidson method to be as fast as the JD method in the case of diagonal preconditioning is no longer applicable; hence the JD method is then expected to be faster in the final phase, too.

Another situation in which one could prefer the JD method is when it is helpful to use inner iterations to compensate for a too poor preconditioning (and thus avoid too many steps of Algorithm 2.1 with frequent shrinking of the basis; see next section). Indeed, the JD method has been explicitly designed to accommodate inner iterations,

and it is not difficult to extend the above analysis to this framework; see [14] for details. Concerning the Davidson method, however, although it is in principle possible to compute $\mathbf{t}_k^{(D)}$ by solving approximately $(A-\theta_k I)\,\mathbf{t}=\mathbf{r}_k$, we would not advise one to do so. Indeed, first this system is hard to solve iteratively, being indefinite and increasingly ill conditioned as θ_k converges to an eigenvalue. Next, the preconditioner implicitly defined in this way is in general not positive definite, and the convergence, therefore, cannot be guaranteed. Moreover, this is precisely the kind of situation for which the method may indeed behave poorly in practice, as observed in [23].

Remark. Our analysis disregards the subspace acceleration present in Algorithm 2.1. Its effects are indeed very difficult to analyze. In practice, we generally observed that they are noticeable in the early stages of the process, especially when the initial approximation is not very good. In particular, subspace acceleration then most often suffices to prevent stagnation during the first phase, when M_k is still indefinite. Concerning the final phase, for which M_k is positive definite (and to which our analysis applies), it is interesting to consider the results in [17], although they do not directly apply to the present framework. Indeed, they suggest an acceleration similar to that achieved by Krylov subspace methods compared to standard steepest descent. The heuristic reasoning in [9] goes along the same lines. It is based on an analogy with the conjugate gradient method (for solving linear systems), considering the ultimate phase for which θ_k has converged and is virtually constant. On this basis, some shrinking strategies are discussed in [9], where it is also proposed to further improve the acceleration algorithm by using refined Ritz vectors whenever appropriate. Now, these developments do not allow a direct comparison between different methods to compute $\mathbf{t}_k$. However, they suggest that working with the whole subspace accelerates the convergence of all methods in essentially the same way, i.e., a method that is faster when working with $\mathrm{span}\{\mathbf{u}_k,\,\mathbf{t}_k\}$ only is expected to remain faster with subspace acceleration. From that point of view, the discussion above keeps all its relevance also in the presence of subspace acceleration.

5. Further numerical results. We now consider a more realistic experiment. The matrix is 1000×1000 and given by

$$a_{ij} \;=\; \begin{cases} i & \text{if } j=i,\\ \alpha & \text{if } 1\le |j-i|\le 10,\\ 0 & \text{otherwise.}\end{cases}$$

We tested Algorithm 2.1 with two initialization strategies: $V_1=[\mathbf{e}_1\ \mathbf{e}_2]$ (*Init*1) and $V_1=[\frac{\mathbf{v}}{\|\mathbf{v}\|}]$ with $\mathbf{v}=0.99\,\mathbf{e}_1+0.01\,\mathbf{e}$ (*Init*2). To define $\mathbf{t}_k$ at substep 7, we considered the following variants:

Davidson: the Davidson method defined by (2.1) and (2.2).

JD($D-\theta_k I$*)*: the simple JD scheme defined by (2.2) and (2.6).

JDCG($D-\theta_k I$*)*: the JD method with inner preconditioned conjugate gradient iterations to solve the correction equation (2.3), using for these inner iterations the projected preconditioner (2.4) with M_k given by (2.2).

JDCG($D-\tau I$*)*: the JD method with inner preconditioned conjugate gradient iterations to solve the correction equation (2.3), using for these inner iterations the projected preconditioner (2.4) with $M_k=\mathrm{diag}(A-\tau I)$, where τ is the largest number such that $A-\tau I$ is (nonstrictly) diagonally dominant (and, therefore, positive definite because it is irreducibly diagonally dominant; see [28, Theorem 1.8]); note that this is a θ_k-independent preconditioner.

TABLE 1
Number of multiplications by A needed to reach $\mathbf{r}_k < 10^{-10}$.

α	1000	100	10	1	0.1	0.01
			$j_{\min} = 1$, $j_{\max} = 2$			
*Init*1:						
Davidson	> 999	> 999	> 999	101	26	17
JD(D−θ_k I)	> 999	> 999	> 999	93	14	7
JDCG(D−θ_k I)	532	193	98	28	12	8
JDCG(D−τ I)	466	184	98	40	15	9
JDCG(SSOR)	240	131	86	36	16	9
*Init*2:						
Davidson	> 999	> 999	991	> 999	> 999	> 999
JD(D−θ_k I)	> 999	> 999	> 999	94	18	11
JDCG(D−τ I)	391	150	93	43	18	11
JDCG(SSOR)	241	119	78	43	19	12
			$j_{\min} = 5$, $j_{\max} = 10$			
*Init*1:						
Davidson	460	141	68	21	9	6
JD(D−θ_k I)	462	141	68	20	9	6
JDCG(D−θ_k I)	296	153	79	27	12	8
JDCG(D−τ I)	313	146	93	40	15	9
JDCG(SSOR)	208	119	75	33	16	9
*Init*2:						
Davidson	453	138	73	82	55	53
JD(D−θ_k I)	451	146	73	27	13	10
JDCG(D−τ I)	298	146	91	43	18	11
JDCG(SSOR)	211	118	78	43	19	12
			$j_{\min} = 25$, $j_{\max} = 50$			
*Init*1:						
Davidson	214	94	39	17	9	6
JD(D−θ_k I)	214	94	40	17	9	6
JDCG(D−θ_k I)	310	153	79	27	12	8
JDCG(D−τ I)	301	146	93	40	15	9
JDCG(SSOR)	207	119	75	33	16	9
*Init*2:						
Davidson	212	95	45	48	49	49
JD(D−θ_k I)	212	98	46	23	13	10
JDCG(D−τ I)	298	146	91	43	18	11
JDCG(SSOR)	212	118	78	43	19	12

JDCG(SSOR): the JD method with inner preconditioned conjugate gradient iterations to solve the correction equation (2.3), using for these inner iterations the projected preconditioner (2.4) with M_k equal to the SSOR preconditioning of $A - \tau I$ with relaxation parameter equal to 1 (see, e.g., [20]); τ is as above the largest number such that $A - \tau I$ is (nonstrictly) diagonally dominant and this preconditioner is also θ_k-independent.

For the last three variants, inner iterations were stopped according to the criteria suggested in [13]. We did not test JDCG($D - \theta_k I$) with the second initialization strategy because M_k might then be indefinite and the present version of the used code [12] does not allow for indefinite preconditioning.

We are interested here in the global convergence behavior, and we report in Table 1 the number of multiplications by A needed to achieve $\|\mathbf{r}_k\| < 10^{-10}$.

As expected from the theory, there is hardly any difference between *Davidson* and *JD(D–θ_k I)* with *Init*1, whereas *JD(D–θ_k I)* is significantly better with *Init* 2 for small α, *Davidson* not being able to benefit from the improvement of the preconditioner. The relative behavior of *JDCG(D$-\theta_k$ I)* is also not surprising. If one can pay for large shrinking parameters $j_{\min}$ and $j_{\max}$, then it is faster to skip inner iterations because the latter combine the search directions in a given way, which cannot compete with the global optimization performed by the Rayleigh–Ritz procedure. On the other hand, frequent shrinking of the basis spoils this global optimization, whereas schemes based on inner conjugate gradient iterations are not deeply affected. The latter may thus allow substantial savings, according to the case at hand (relative cost of a multiplication by A and available memory).

Comparing the results for *JDCG(D$-\theta_k$ I)* with those for *JDCG(D$-\tau$ I)*, this experiment also confirms that JDCG is on the whole as efficient with a θ_k-independent preconditioner. This is good news. Indeed, when the matrix is not nearly diagonal, trying to improve the preconditioner while keeping the θ_k-dependent form (2.7) is not easy because indefiniteness may occur and because only relatively small preprocessing cost is affordable since the preconditioner changes at each step. On the other hand, plenty of methods are available to precondition $A-\tau I$, especially if τ is such that the latter matrix is positive definite. Our example was perhaps not very well chosen in this respect because the band structure of the matrix entails that a mere incomplete LU factorization of $A-\tau I$ actually delivers an exact Cholesky factorization (see, e.g., [20]). It then would not have been fair to report the results obtained in such a particular case as representative of the potentialities of this preconditioning method for general matrices. We, therefore, confined ourselves to SSOR preconditioning, which may be seen as an intermediate step between diagonal and incomplete LU preconditioning. The results obtained with *JDCG(SSOR)* illustrate anyway that, in hard situations, the best thing to do is try to improve the preconditioner, significant savings being possible if one is successful.

6. Conclusions. When one considers simple diagonal preconditioning and initializes the method in the standard way (with few canonical vectors as starting subspace), only a little difference may be seen between the Davidson and JD methods. In particular, one need not fear poor behavior of the Davidson method because the matrix would be too close to a diagonal one.

On the other hand, the JD method may bring a significant improvement for matrices not close to a diagonal one, so that the diagonal preconditioning is too poor. Indeed, one would then like to improve the situation by considering either inner iterations or more general (nondiagonal) preconditioning (or both). In either case, the JD method appears better suited, as it has been explicitly designed to accommodate inner iterations and to work with any type of preconditioning.

Appendix: Proof of (4.21). Using the notation introduced at the beginning of the proof of Theorem 4.2, and remembering that $0 \le |\eta|, \delta \le 1$, (4.19) gives, neglecting $\mathcal{O}(\varepsilon)$ terms,

$$\|M_k^{-1}\,\mathbf{r}_k\| \;=\; \frac{\delta\,|\eta-\delta|}{1-\delta^2} \;\le\; \frac{2\,\delta}{1-\delta^2},$$

whereas (4.20) yields (since $(1+\eta^2-2\,\eta\,\delta) \geq (1+\eta^2\,\delta^2-2\,\eta\,\delta) \geq (1-\delta)^2$)

$$\begin{aligned}\|\widetilde{\mathbf{t}}_k^{(D)}\| &\geq \frac{\left(\widetilde{\mathbf{t}}_k^{(D)},\,(A-\theta_k\,I)\,\widetilde{\mathbf{t}}_k^{(D)}\right)^{1/2}}{\sqrt{a_{nn}-a_{11}}} \\ &= \frac{\varepsilon\,\delta}{1-\delta^2}\;\left((\eta-\delta)^2\,(1-\eta^2)+(1+\eta^2-2\,\eta\,\delta)^2\right)^{1/2}\;\sqrt{\frac{\beta}{a_{nn}-a_{11}}} \\ &\geq \frac{\varepsilon\,\delta}{1-\delta^2}\;(1-\delta)^2\;\left(\sum_{j=2}^{n}\frac{b_{1j}^2}{(a_{jj}-a_{11})(a_{nn}-a_{11})}\right)^{1/2}.\end{aligned}$$

Inequality (4.21) then readily follows.

Acknowledgments. Both referees are deeply acknowledged for their careful reading and for several suggestions that helped to improve the manuscript.

REFERENCES

[1] O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems: Theory and Computation*, Classics Appl. Math. 35, SIAM, Philadelphia, 2001.

[2] M. Crouzeix, B. Philippe, and M. Sadkane, *The Davidson method*, SIAM J. Sci. Comput., 15 (1994), pp. 62–76.

[3] 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.

[4] 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.

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

[6] A. V. Knyazev and K. Neymeyr, *A geometric theory for preconditioned inverse iteration*. III: *A short and sharp convergence estimate for generalized eigenvalue problems*, Linear Algebra Appl., 358 (2003), pp. 95–114.

[7] R. B. Morgan, *Preconditioning eigenvalues and some comparison of solvers*, J. Comput. Appl. Math., 123 (2000), pp. 101–115.

[8] R. B. Morgan and D. S. Scott, *Generalizations of Davidson's method for computing eigenvalues of sparse symmetric matrices*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 817–825.

[9] C. W. Murray, S. C. Racine, and E. R. Davidson, *Improved algorithms for the lowest few eigenvalues and associated eigenvectors of large matrices*, J. Comput. Phys., 103 (1992), pp. 382–389.

[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] K. Neymeyr, *A geometric theory for preconditioned inverse iteration* II: *Convergence estimates*, Linear Algebra Appl., 322 (2001), pp. 87–104.

[12] Y. Notay, *JDCG, a Matlab Package for the Computation of a Few of the Smallest Eigenpairs of a Real Symmetric Matrix*, http://homepages.ulb.ac.be/∼ynotay, 2004.

[13] Y. Notay, *Combination of Jacobi-Davidson and conjugate gradients for the partial symmetric eigenproblem*, Numer. Linear Algebra Appl., 9 (2002), pp. 21–44.

[14] Y. Notay, *Convergence analysis of inexact Rayleigh quotient iteration*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 627–644.

[15] S. Oliveira, *On the convergence rate of a preconditioned subspace eigensolver*, Computing, 63 (1999), pp. 219–231.

[16] E. Ovtchinnikov, *Convergence estimates for the generalized Davidson method for symmetric eigenvalue problems* I: *The preconditioning aspect*, SIAM J. Numer. Anal., 41 (2003), pp. 258–271.

[17] E. Ovtchinnikov, *Convergence estimates for the generalized Davidson method for symmetric eigenvalue problems* II: *The subspace acceleration*, SIAM J. Numer. Anal., 41 (2003), pp. 272–286.

[18] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice–Hall, Englewood Cliffs, NJ, 1980.

[19] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Halstead Press, New York, 1992.
[20] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, New York, 1996.
[21] V. Simoncini and L. Eldén, *Inexact Rayleigh quotient-type methods for eigenvalue computations*, BIT, 42 (2002), pp. 159–182.
[22] G. Sleijpen, A. Booten, D. Fokkema, and H. van der Vorst, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems*, BIT, 36 (1996), pp. 595–633.
[23] 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.
[24] G. Sleijpen, H. van der Vorst, and E. Meijerink, *Efficient expansion of subspaces in the Jacobi-Davidson method for standard and generalized eigenproblems*, Electron. Trans. Numer. Anal., 7 (1998), pp. 75–89.
[25] G. L. G. Sleijpen and F. W. Wubs, *Exploiting multilevel preconditioning techniques in eigenvalue computations*, SIAM J. Sci. Comput., 25 (2003), pp. 1249–1272.
[26] A. Stathopoulos and C. F. Fischer, *A Davidson program for finding a few selected extreme eigenpairs of a large, sparse, real, symmetric matrix*, Comput. Phys. Comm., 79 (1994), pp. 268–290.
[27] J. van den Eshof, *The convergence of Jacobi–Davidson iterations for Hermitian eigenproblems*, Numer. Linear Algebra Appl., 9 (2002), pp. 163–179.
[28] R. S. Varga, *Matrix Iterative Analysis*, Prentice–Hall, Englewood Cliffs, NJ, 1962.
[29] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 2, pp. 544–565

TASK SCHEDULING IN AN ASYNCHRONOUS DISTRIBUTED MEMORY MULTIFRONTAL SOLVER*

PATRICK R. AMESTOY†, IAIN S. DUFF‡, AND CHRISTOF VÖMEL§

Abstract. We describe the improvements to the task scheduling for MUMPS, an asynchronous distributed memory direct solver for sparse linear systems. In the new approach, we determine, during the analysis of the matrix, candidate processes for the tasks that will be dynamically scheduled during the subsequent factorization. This approach significantly improves the scalability of the solver in terms of execution time and storage. By comparison with the previous version of MUMPS, we demonstrate the efficiency and the scalability of the new algorithm on up to 512 processors. Our test cases include matrices from regular three-dimensional grids and irregular grids from real-life applications.

Key words. sparse linear systems, high performance computing, MUMPS, multifrontal Gaussian elimination, distributed memory code, task scheduling

AMS subject classifications. 65F05, 65F35, 65F50

DOI. 10.1137/S0895479802419877

1. Introduction.

We consider the direct solution of sparse linear systems on distributed memory computers. Two state-of-the-art codes for this task, MUMPS[1] and SuperLU, have been extensively studied and compared in [5]. Specifically, the authors show that on a large number of processors, the scalability of the multifrontal approach used by MUMPS [4, 5] with respect to computation time and use of memory could be improved. This observation is the starting point for this current work.

The solution of a linear system of equations using MUMPS consists of three phases. In the *analysis* phase, the matrix structure is analyzed and a suitable ordering and data structures for an efficient factorization are produced. In the subsequent *factorization* phase, the numerical factorization is performed. The final *solve* phase computes the solution of the system by forward and backward substitution.

The numerical factorization is the most expensive of these three phases, and we now describe how parallelism is exploited in this phase. The task dependency graph of the multifrontal factorization is a tree, the so-called *assembly tree*. A node of this tree corresponds to the factorization of a dense submatrix, and an edge from one node to another describes the order in which the corresponding submatrices can be factorized. In particular, independent branches of the assembly tree can be factorized in parallel. Moreover, each node in the tree can further expose parallelism opportunities. The ScaLAPACK library [8] provides an efficient parallel factorization of dense matrices and is used for the matrix associated with the root of the assembly tree. But MUMPS offers another possibility for exploiting parallelism for those nodes that are large enough. Such nodes can be assigned a master process during analysis that chooses, during numerical factorization, a set of slave processes to work on subblocks of the

*Received by the editors December 13, 2002; accepted for publication (in revised form) by E. Ng April 19, 2004; published electronically January 12, 2005.
http://www.siam.org/journals/simax/26-2/41987.html
†ENSEEIHT, 2 rue Camichel, BP 7122, 31071 Toulouse Cedex 7, France (amestoy@enseeiht.fr).
‡CERFACS, Toulouse, and Atlas Centre, RAL, Oxon OX11 0QX, England (I.Duff@rl.ac.uk).
§CERFACS, 42 ave G. Coriolis, 31057 Toulouse Cedex, France (Christof.Voemel@cerfacs.fr).
[1]The MUMPS package is available at http://www.enseeiht.fr/apo/mumps.

dense matrix. This *dynamic* decision about the slaves is based on the load of the other processors; only the less loaded ones are selected to participate as slaves.

In order to address the scalability issues, we have modified this task scheduling and the treatment of the assembly tree during analysis and factorization. We now give a brief description of these new modifications to Version 4.1 of MUMPS (to which we sometimes refer as the old code or the previous version of MUMPS).

The objective of the dynamic task scheduling is to balance the work load of the processors at run time. In the previous version of MUMPS, a master process is free to choose its slaves from among all available processes. Since this choice is taken dynamically during the factorization phase, we have to anticipate it by providing enough memory on every process for the corresponding computational tasks. Since typically not all processes are actually used as slaves (and, on a large number of processors, often only relatively few are needed), the prediction of the required workspace can be severely overestimated. Second, decisions concerning a node should take account of global information on the assembly tree to localize communication.

With the concept of *candidate processors*, it is possible to guide the dynamic task scheduling and to address these issues. The concept originates in an algorithm presented in [28, 29] and has also been used in the context of static task scheduling for sparse Cholesky factorization [19]. In this paper, we show how it also extends efficiently to dynamic scheduling. For each node that requires slaves to be chosen dynamically during the factorization, we introduce a limited set of processors from which the slaves can be selected. This allows us to exclude all noncandidates from the estimation of workspace during the analysis phase and leads to a more realistic prediction of the workspace needed. Furthermore, the candidate concept allows us to better structure the computation since we can explicitly restrict the choice of the slaves to a certain group of processors and enforce, for example, a "subtree-to-subcube" mapping principle (see [17]). (Throughout this paper, we assume that every processor has one single message passing interface (MPI) process associated with it so that we can unambiguously identify a processor and a corresponding MPI process.)

We illustrate the benefits of the new approach by tests using a number of performance metrics, including execution time, memory usage, communication volume, and scalability. Our results demonstrate significant improvements for all these metrics, in particular when performing the calculations on a large number of processors.

The rest of this paper is organized as follows. In section 2, we review briefly the general concepts of the multifrontal direct solution of sparse linear systems. We describe in section 3 the possibilities for exploiting parallelism. We then introduce, in section 4, the concept of candidate processors. In section 5, we give an overview of how the candidate concept fits into the scheduling algorithm, and we present the algorithmic details in section 6. Section 7 gives an overview of the test problems used in this paper. The presentation of our experimental results begins with parameter studies and detailed investigations of the improved algorithms in section 8. Afterward, we present a systematic comparison of the previous version with the new version of the code on regular grid problems and general matrices in section 9. Finally, we discuss possible extensions of our algorithm in section 10 and present our conclusions and a brief summary in section 11.

2. Tasks and task dependencies in the multifrontal factorization. We consider the direct solution of large sparse systems of linear equations $Ax = b$ on distributed memory parallel computers using multifrontal Gaussian elimination. For an unsymmetric matrix, we compute its LU factorization; if the matrix is symmetric,

its LDL^T factorization is computed.

The multifrontal method was initially developed for indefinite sparse symmetric linear systems [13] and was then extended to unsymmetric matrices [14]. We limit our attention to general unsymmetric and symmetric indefinite matrices in the following, but for an overview of the multifrontal method for symmetric positive definite systems we refer to [11, 13, 23].

In this section, we describe the tasks arising in the factorization phase of a multifrontal algorithm. Specifically, we investigate the work associated with the factorization of individual frontal matrices and the order in which these factorizations can be performed.

The so-called elimination tree [13, 22] 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 eliminated. For a general sparse matrix, the definition yields a *partial* ordering which allows some freedom for the sequence in which pivots can be eliminated. One central concept of the multifrontal approach [13] is to group (or *amalgamate*) columns with the same sparsity structure to create bigger *supervariables* or *supernodes* [13, 24] in order to make use of efficient dense matrix kernels. It is common to relax the criterion for amalgamation and permit the creation of coarser supernodes with extra fill-in that, however, improve the performance of the factorization (see [7, 13]). The amalgamated elimination tree is called the *assembly tree*.

Frontal matrices are always considered as dense matrices to allow us to use efficient BLAS kernels and avoid indirect addressing (see, for example, [10]). Within the frontal matrix, pivots are eliminated only in the block of so-called fully summed variables. Afterwards, the *contribution block* of the node, i.e., the Schur complement matrix, is computed and used to update the rows and columns of the overall matrix which are associated with its parent node. The parent node assembles the contribution blocks from all its children nodes into its own frontal matrix, and the elimination process is then performed on the parent.

3. Parallelism in the multifrontal factorization. In the following, we identify different sources of parallelism in the multifrontal factorization and describe how these are exploited in MUMPS [4].

3.1. The different types of parallelism. In section 2, we mentioned that the tasks of multifrontal Gaussian elimination for sparse matrices are only *partially* ordered. 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. Experimental results presented in [3] showed only a limited speedup from tree parallelism. On the other hand, often more than 75% of the computations are performed in the top three levels of the assembly tree [2]. For better scalability, additional parallelism is created from blocked factorization algorithms.

The computation of the Schur complement of frontal matrices with a large enough contribution block can be performed in parallel using a master-slave computational model. The contribution block is partitioned and each part of it assigned to a slave. The master processor is responsible for the factorization of the block of fully summed variables and sends the triangular factors to the slave processors which then update their own share of the contribution block independently from each other and in parallel. We refer to this approach as *type* 2 *parallelism* and call the concerned nodes *type* 2 *nodes*.

Furthermore, the factorization of the dense root node can be treated in parallel with ScaLAPACK [8]. The root node is partitioned and distributed to the processors using a two-dimensional (2D) block cyclic distribution. This is referred to as *type* 3 *parallelism*.

MUMPS performs the factorization of the pivot rows of a frontal matrix on a single processor which can sometimes lead to performance problems. For this reason, it is possible to create artificial type 2 parallelism by splitting the pivot block [4].

3.2. Parallel task scheduling: Main principles. We describe in this section the techniques implemented in Version 4.1 of MUMPS [3, 4] which has been extensively tested and compared with SuperLU [5] and WSSMP [18]. We also present the proportional mapping by Pothen and Sun [28, 29] from which we develop, in section 4, our idea of the candidate-based scheduling that is used in the new version of MUMPS.

3.2.1. Geist–Ng mapping and layers in the assembly tree. Our previous scheduling approach consists of two phases. At first, we find the lower part of the assembly tree where enough tree parallelism can be obtained. Afterward we process the remaining upper part of the tree, exploiting additionally type 2 and type 3 parallelism.

The mapping algorithm by Geist and Ng [15] allows us to find a layer in the assembly tree so that the subtrees rooted at the nodes of this layer can be mapped onto the processors for a good balance with respect to floating-point operations. We call the constructed layer L_0. We then recursively define the following layer partition. Given a node in layer L_{i-1}, the parent of this node belongs to L_i if and only if all the children of this parent node belong to the layers $L_0, \ldots, L_{i-1}$. As the nodes in one layer can be processed only if all their children, belonging to the lower layers, have already been treated, the layer partition represents not only dependency but also concurrency of the multifrontal factorization. An example is shown in Figure 3.1.

3.2.2. The proportional mapping of Pothen and Sun. The proportional mapping approach by Pothen and Sun [28, 29] represents an alternative approach to task scheduling in both regular and possibly irregular assembly trees.

The assembly tree is processed from top to bottom, starting with the root nodes. For each root node, we calculate the work associated with the factorization of all nodes in its subtree, and the available processors are distributed among the root nodes according to their weight. Each node thus gets its set of *preferential* processors. The same partitioning is now repeated recursively. The processors that have been previously assigned to a node are now distributed among the children proportionally to their weight (as given by the computational costs of their subtrees). The recursive partitioning stops once a subtree has only one processor assigned to it.

The proportional mapping (illustrated in Figure 3.2) both achieves locality of communication and guides the partitioning from a *global* point of view, taking account of the weight of subtrees.

3.2.3. Dynamic task scheduling for type 2 parallelism. The static task mapping is performed on the basis of *estimated* costs of computational work which is inaccurate, in particular if pivots have to be delayed for numerical reasons. For a better equilibration of the actual computational work at run time, both the number and the choice of the slaves of type 2 nodes are *dynamically* determined during factorization [3, 4]. When the master of a type 2 node receives the symbolic information on the structure of the contribution blocks of the children, the slaves for the factorization are selected based on their current work load, the least loaded processors being chosen.

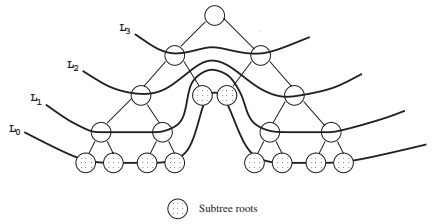


FIG. 3.1. *Layers in the assembly tree.*

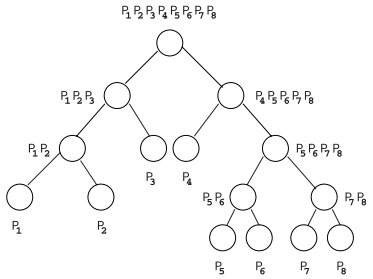


FIG. 3.2. *Proportional mapping of an assembly tree on eight processors.*

In MUMPS, if a node above layer L_0 possesses a contribution block larger than a given threshold and the number of eliminated variables in its pivot block is large enough, then it becomes a type 2 node. In the new version of MUMPS, we restrict the freedom for the dynamic choice of the slaves to the candidates that have been chosen for a given node during the analysis phase. This is explained in detail in section 4.2.

4. Combining the concept of candidates with dynamic task scheduling. In this section, we first give a more detailed illustration of the shortcomings of dynamic scheduling on a large number of processors and then propose as a solution an algorithm that exploits the concept of candidate processors.

4.1. Issues of dynamic scheduling. In MUMPS, the amount of memory needed for each processor is estimated during the analysis phase and is reserved as workspace for the factorization. Consequently, if every processor can possibly be taken as a slave of a type 2 node, then enough workspace has to be reserved, on each processor, during the analysis phase for the potential corresponding computational task. This can lead to a dramatic overestimate of memory requirements because, during the factorization, typically not all processors are actually used as slaves.

Second, the choice of the slaves is completely local. When a type 2 node is to be processed, its master greedily takes the slaves that seem best to it; those processors that are less loaded (with respect to the number of floating-point operations) than itself at the time of the scheduling decision are selected as slaves. Thus, the decision about the slaves depends crucially on the instant when the master chooses the slaves (locality in time). Furthermore, no account is taken of other type 2 nodes in the tree that have to be processed (locality in space). Instead of sharing the available slaves so that other nodes can be processed in parallel, a master might decide to take all of them, hindering the work on other branches of the assembly tree.

4.2. Candidate processors for type 2 parallel nodes. In the following, we present a concept of *candidate processors* that naturally addresses the issues raised in section 4.1. For each type 2 node that requires slaves to be chosen dynamically during the factorization because of the size of its contribution block, we introduce a limited set of processors from which the slaves can be selected. While the master previously chose slaves from among all less loaded processors, slaves are now only chosen from this list of candidates. This effectively allows us to exclude all noncandidates from the estimation of workspace during the analysis phase and leads to a tighter and more realistic estimation of the workspace needed. Second, we can expect a performance gain in cases as described in the previous section where greedy decisions of one type 2

master can no longer hinder processors from processing another node.

The candidate concept can be thought of as an intermediate step between full static and full dynamic scheduling. While we leave some freedom for dynamic decisions at run time, this is guided by static decisions about the candidate assignment during the analysis phase. We refer to section 6.6 for a full description of the algorithmic details.

The assignment and the choice of the candidate processors is guided using a proportional mapping as described in sections 3.2.2 and 5.1. We partition the set of processors recursively, starting from the root, so that for each subtree there is a well-defined subset of preferential processors from which the candidates can be selected.

5. Task mapping and task scheduling in MUMPS. In this section, we discuss in general terms our improvements to MUMPS as they have been integrated into the new Version 4.2. With task *mapping*, we refer to the assignment of master processors and candidates during the analysis phase, and with task *scheduling* to the dynamic choice of type 2 slaves during the factorization phase.

5.1. Task mapping algorithm during the analysis phase. A first major point to emphasize is the greater flexibility and adaptivity of the new algorithm when mapping the upper part of the assembly tree (that is, above layer L_0). The former version, Algorithm 1, performs a simple mapping of only the master nodes, while the new version, Algorithm 2, treats the upper part layerwise, mapping both master nodes and type 2 candidates.

The second contribution of the new algorithm is of course the added features. A very important feature is the candidate concept guided by a proportional mapping partition of the processors. Furthermore, we have added to the treatment of each layer a preprocessing step that performs amalgamations and node splitting. Moreover, we have improved the construction of layer L_0 for better memory scalability. Lastly, we treat memory imbalances due to type 2 node mapping using a postprocessing step.

Algorithm 1 Old task mapping algorithm.

(1) Given the assembly tree of a sparse matrix A
(2) Build and map initial layer L_0
(3) Decide type of parallelism for nodes in upper part of tree
(4) Map master nodes of upper part of tree

The starting point (1) of the original algorithm is the assembly tree that was constructed from the elimination tree of a given sparse matrix. From this assembly tree, the algorithm constructs, in step (2), an initial layer L_0 following the Geist–Ng approach (section 3.2.1). Afterward, it is decided for which nodes type 2 or type 3 parallelism is exploited (3), and finally the masters of all nodes above layer L_0 are mapped (4) with the objective of balancing the memory.

The starting point (1′) of the new algorithm is the same assembly tree as for the old approach (1). Step (2′) differs from the corresponding step (2) in the old algorithm insofar as the constructed initial layer controls better the memory demands of the subtree roots; see section 6.1 for the details. In step (3′), we calculate a variant of the proportional mapping whose algorithmic description is given in section 6.2. For each node in the assembly tree, we obtain a set of preferential processors that will guide the selection and mapping of the candidate processors in step (6′). Step (4′) performs amalgamations and node splitting to improve the nodes of the current layer. In step (5′), we decide which type of parallelism we exploit for the nodes of the

Algorithm 2 New task mapping algorithm.

```
(1′) Given the assembly tree of a sparse matrix A
(2′) Build and map modified initial layer L_0
(3′) Calculate relaxed proportional mapping, i.e. the preferential processors
current_layer = 1
while there exist unmapped nodes on or above current_layer do
  (4′) Perform tree modifications if necessary
  (5′) Decide type of parallelism for the nodes on current_layer
  (6′) Map the tasks associated with the nodes on current_layer
  current_layer = current_layer + 1
end while
(7′) Postprocessing of the candidate selection to improve memory balance
```

current layer. The list of tasks associated with the current layer includes the masters for the type 1 and type 2 nodes, and the type 2 candidates which are derived from the proportional mapping (2′); see section 6.2. For the task mapping, we use a list scheduling algorithm that is described in section 6.4. The main difference between the new mapping (6′) and the old one (4) is that we now preassign candidate processors for the type 2 nodes. The postprocessing step (7′) intends to improve the memory balance through remapping of the type 2 masters; see section 6.5.

5.2. Task scheduling during the factorization phase. In this section, we describe in Algorithm 3 the task management of a processor during the factorization phase.

Algorithm 3 Dynamic task scheduling performed on a processor during factorization.

```
(1) Given the task pool of one processor
while (2) Not all tasks processed do
  if Work is received from another processor then
    (3) Store work in pool of tasks
  else
    (4) Extract work from the task pool
    if Task is master of type 2 node then
      (5) Choose and notify the slaves for the type 2 node
    end if
    (6) Perform pivot elimination and/or contribution block update
  end if
end while
```

The task pool (1) of a processor can contain the following tasks: master of a type 1 node, master of a type 2 node, or slave of a type 2 node. The processor adds new tasks (3) or extracts them from the pool (4), respectively. If, during the factorization, the task pool of the processor is empty, it will wait until it receives new tasks and then re-enter loop (2). If the processor works as a type 2 master, it chooses the slaves that will participate in the parallel contribution block update (5) before it starts the elimination of the pivotal block (6). Otherwise, if the processor is a type 1 master or a type 2 slave, it begins directly with the pivot elimination or the contribution block update, respectively, (6).

In the new version of the algorithm, only step (5) is modified to ensure that the type 2 slaves are selected from among the candidates allocated for the type 2 node. We give the details of the algorithm for choosing the slaves in section 6.6.

6. Details of the improved task mapping and scheduling algorithms. After the general comparison of the old and new versions of MUMPS task mapping

and scheduling in section 5, we describe in this section the key points of the new algorithm in detail.

6.1. The Geist–Ng construction of layer L_0. We now describe the construction of the initial layer L_0 that extends the Geist–Ng approach from section 3.2.1.

Algorithm 4 The Geist–Ng algorithm.

(1) Let L_0 contain all root nodes of the assembly tree
(2) Map layer L_0
while (3) Layer L_0 is not acceptable **do**
  (4) Find node in L_0 with highest computational costs
  (5) Replace this node by its children in L_0
  (6) Map new layer L_0
end while

Starting with a potential layer L_0 consisting of the root nodes of the assembly tree (1), we first compute (2) a mapping of L_0 with the list scheduling heuristics described in section 6.4. The former criterion for accepting the layer in step (3) demands that the load imbalance between the processors is smaller than a threshold. Here, the work associated with a node in L_0 is defined as the costs for computing the factors of the subtree rooted at the node and can be estimated during the analysis phase. If the mapping of layer L_0 is not acceptable, then the node with the highest costs is eliminated from the layer and replaced by its children (4, 5). A new mapping is computed (6) with the same algorithm as in (2).

The main problem of Algorithm 4 is that balancing the computational work does not necessarily imply a good memory balance, in particular if nodes with a very small number of pivots but a big contribution block have to be mapped. For better memory balance, we modify the criterion of acceptability (3) to demand that both the load imbalance for the mapping of L_0 is smaller than a threshold *and* that L_0 contains no nodes that would need to be amalgamated.

6.2. The relaxed proportional mapping. Algorithm 5 describes one step of the proportional mapping presented in section 3.2.2. The preferential processors given to a node are distributed among its children according to their weight. Note that we can *relax* the strict proportional mapping by multiplying the number of preferential processors n_a by a relaxation factor $\rho \geq 1$ in step (2).

Algorithm 5 One step of proportional mapping.

Given a node n with preferential processors $p_1, \ldots, p_{n_a(n)}$ and children $s_1, \ldots, s_i$
for each child s of n **do**
  (1) Calculate relative costs $c_r(s)$ of child s, $0 \leq c_r(s) \leq 1$
  (2) Calculate number of preferentials $n_a(s) = \min\{\rho \times c_r(s) \times n_a(n), n_a(n)\}$ for child s
end for
(3) Cyclic assignment of the preferential processors for all children $s_1, \ldots, s_i$

In step (1), we calculate the relative costs $c_r(s)$ of a child $s, s \in \{s_1, \ldots, s_i\}$ from the costs $c(s)$ for the factorization of all nodes in the subtree rooted at s as $c_r(s) = c(s)/\sum_{k=1}^{i} c(s_k)$. From the relative weight $c_r(s)$ of child s, we obtain its share of preferential processors in step (2) that can be relaxed by the factor ρ. After we have calculated the number of preferential processors for all children, we distribute in step (3) the processors $p_1, \ldots, p_{n_a(n)}$ among the children.

6.3. Choosing the number of candidates for a type 2 node. Our approach consists of two steps. For a given layer, we first determine for each type 2 node the number n_c of candidate processors. In a second step, we choose the candidates from the available processors. As the selection of a candidate processor is conceptually similar to the selection of the master processors for the type 1 and type 2 nodes, we hope to obtain better load balancing by mapping the master and candidate processors together; see section 6.4.

We have experimented with two different ways for determining the number of candidates for a given type 2 node and describe these in Algorithm 6. In the first approach, we select its preferential processors as candidates, thus setting the number of candidates equal to the number of preferentials. In a second approach, we employ an additional postprocessing step, where we redistribute the candidates of the layer according to the relative weight of the nodes. As the proportional mapping is calculated from the costs of complete subtrees, not individual nodes, a large node on a given layer might have only a relatively small number of preferentials. For this reason we can reassign candidates on the same layer by the optional step in Algorithm 6.

Algorithm 6 Determining the number of candidates using the preferentials.

Given a layer in the assembly tree
for each Type 2 node n with $n_a(n)$ preferential processors in the layer **do**
  (1) Determine the number of candidates by $n_c(n) = n_a(n)$.
end for
(2) OPTIONAL: Redistribute the total number of candidates of the layer among the layer's type 2 nodes according to their relative weight.

6.4. Layerwise task mapping. We use for the task mapping a variant of the well-known list scheduling algorithm [20]. We first make a list of the tasks sorted by decreasing costs, and then we map the tasks in this order one after another to the processor that has the least work assigned so far.

In the case of layer L_0, we employ the original list scheduling [20]; however, for all upper layers $L_1, L_2, \ldots$ our algorithm is more complicated for two reasons. First, we want to guide mapping decisions by the proportional mapping representing a global view of the tree. Second, we have to take care of constraints that arise either from explicit user-given limits on memory or work for each processor, or implicitly from the fact that any two candidate processors or any candidate and the master of a type 2 node have to be different from each other.

Algorithm 7 Generic mapping algorithm.

(1) Create an ordered task list
while Task list not empty **do**
  (2) Extract the next task t_i from the list
  (3) Make a preference list for the processors
  while Task t_i not mapped to a processor **do**
    (4) Try to map t_i to the next processor from the preference list
  end while
end while

The first two steps (1) and (2) of Algorithm 7 are identical to the original list scheduling approach: We create a list of all tasks that have to be mapped in the layer, that is, the work of the type 1 node masters, of type 2 node masters, and of type 2 node candidates. This list is then ordered by decreasing costs and the tasks are mapped in the order that they appear in the list.

Steps (3) and (4) are the generalization of the idea of mapping to the least loaded processor. We create a *preference* list containing all the processors, at first the preferential ones ordered by increasing work load and then the nonpreferential ones ordered separately, also by increasing work load. The first processor in the preference list that does not violate the mapping constraints will be the one to which the task is mapped.

6.5. Postprocessing of the assembly tree for an improved memory balance in the *LU* factorization. There is an important difference between symmetric and unsymmetric factorization with respect to memory. In the LDL^T factorization, the master of a type 2 node only holds the pivotal block, whereas, in the LU factorization, the master stores the *complete* fully summed rows. Thus, in the case of the LU factorization, the work equilibration can lead to memory imbalances if the same processor becomes master of several type 2 nodes.

For this reason, after the whole tree is mapped with the objective of balancing the work, we use a postprocessing step to correct memory problems described by Algorithm 8. We process the upper part of the assembly tree from the top down (1), as the type 2 nodes creating the biggest problems are often near a root node. By swapping a master processor with one of the candidates, we locally improve the memory imbalance (steps 2 and 3).

Algorithm 8 Postprocessing for better memory equilibration in the LU factorization.

(1) Process the type 2 nodes in the tree from the root downwards
(2) For a node n with master $p^M(n)$ select candidate $c^*(n)$ with smallest memory
if memory imbalance can be improved by swapping $p^M(n)$ and $c^*(n)$ **then**
  (3) Exchange the roles of master and candidate processor $p^M(n) \Leftrightarrow c^*(n)$
end if

6.6. The dynamic scheduling algorithm used at run time. We show how the candidate concept influences the original scheduling algorithm used in MUMPS and describe the role of the algorithmic parameter $k_{\max}$ controlling the minimum granularity for type 2 parallelism at run time.

Algorithm 9 Dynamic choice of the slaves of a type 2 node.

Given a type 2 node n with master processors $p^M(n)$ and children $s_1, \ldots, s_i$
(1) The masters of the children $p^M(s_1), \ldots, p^M(s_i)$ send symbolic data to $p^M(n)$
(2) $p^M(n)$ analyzes its information concerning the load of all processors
(3) $p^M(n)$ decides the partitioning of the frontal matrix of node n and chooses the slave processors $p_1^S(n), \ldots, p_j^S(n)$
(4) $p^M(n)$ informs all processors working on the children about the partition
(5) The numerical data is sent directly to the slaves $p_1^S(n), \ldots, p_j^S(n)$

In a two-phase assembly process (see Algorithm 9), the master receives the integer data describing the symbolic structure of the front (1) and analyzes the information on the work load of the other processors (2). At step (3), the master processor $p^M(n)$ selects the least loaded among all processors as slaves.

The number of slaves, n_s, for a type 2 node must satisfy the *minimum granularity condition* $n_s \geq \max(\lfloor \frac{ncb}{k_{\max}} \rfloor, 1)$, where ncb denotes the number of rows in the contribution block. The parameter $k_{\max}$ controls the maximum work of a type 2 slave and thus the maximum buffer size permitted for the factorization of a type 2 node.

Once the slaves participating in the parallel update of the contribution block have been selected, they obtain the part of the symbolic information from the master

$p^M(n)$ that is relevant for their work (4). Furthermore, they receive the corresponding numerical data from the processors working on the children (5).

In the candidate-based scheduling approach, we modify step (3) so that the slaves are *always* chosen among the candidates provided for the node. At first, we select all those candidates that are less loaded than the master processor. If minimum granularity is not satisfied, additional candidates are chosen so that it is.

6.7. Complexity of the new mapping algorithm. In this section, we comment on the complexity of our mapping algorithm. We assume we have p processors. We denote by d the maximum distance between layer L_0 and a root node, by n_u the number of nodes above L_0, and by ms the maximum number of sons of a node above layer L_0.

The Geist–Ng construction of layer L_0 using Algorithm 4 replaces a node with its children $\mathcal{O}(n_u)$ times. The corresponding amount of data stored in a linked list is also $\mathcal{O}(n_u)$. We assume that the list is ordered with the greatest work load at its head. Replacing this node with its children is done with a merge sort of the already ordered list excluding the head node but including the list of the children. The overall complexity of step (5) is then $\mathcal{O}((ms * \log(ms) + |L_0|) * n_u)$. The mapping of the list nodes done by Algorithm 7 is then simpler since it does not include step (1) of the algorithm (ordering of the task list).

The proportional mapping from Algorithm 5 is only calculated from the root nodes down to layer L_0. For this reason, the recursion is bounded by the distance d between root node(s) and layer L_0. The amount of data stored is of order $\mathcal{O}(p * n_u)$. For each layer of the assembly tree, a redistribution step is performed. The amount of computation involved is of order $\mathcal{O}(d * n_u/d) = \mathcal{O}(n_u)$, where n_u/d denotes the average number of nodes per layer. We remark that the computational complexity of this step is similar to the optional redistribution step from Algorithm 6.

The mapping algorithm (Algorithm 7) is of central importance for the overall scheme and depends crucially on the efficient construction of the task list. The ordering of the task list is computed by a merge sort which has a complexity of order $\mathcal{O}(n * \log(n))$, where n denotes the number of nodes in the list. Furthermore, at each step, it involves a search for the least loaded among p processors. Thus, the overall complexity is given by $\mathcal{O}(p * n + n * \log(n))$. When constructing layer L_0, the construction of the ordered task list is in fact done at step (5) of Algorithm 4. The overall complexity of Algorithm 4 is thus $\mathcal{O}((ms * \log(ms) + |L_0|) * n_u + n_u * p * |L_0|)$.

Finally, the postprocessing step from Algorithm 8 is performed for the type 2 nodes in the upper part of the tree, for which we have to find the candidate processor with the smallest memory. The cost for this step is thus of order $\mathcal{O}(n_u * p)$.

Our tests with the regular grid problems from section 7 show that in the case of a nested dissection ordering, the layer L_0 typically consists of between p and $2p$ nodes. However, the structure of the layer greatly depends on the shape of the assembly tree which is influenced by the matrix ordering. In practice, for large grid problems, $d = p/4$ is an upper bound (node splitting in general increases the expected bound $\log(p)$), the number of nodes n_u above L_0 usually does not exceed $3p/2$, and ms is typically less than 3.

7. The test environment. In this section, we present the test matrices that we use to illustrate the behavior of our algorithm.

For our tests, we use a CRAY T3E-900 (512 processors, 256 megabytes RAM and 900 peak megaflops per processor), an SGI Origin 2000 (32 processors, 16 gigabytes shared memory, 500 peak megaflops per processor), and an IBM SP3 (29 SMP

nodes with 16 processors and 16 gigabyte memory, 375 megahertz). We consider different orderings including nested dissection from SPARSPAK [16], METIS [21], and SCOTCH [26, 27], and Approximate Minimum Fill [25, 30].

7.1. Regular grid test problems. We consider a set of test matrices obtained from an 11-point discretization of the Laplacian on three-dimensional (3D) grids of either cubic or rectangular shape, the respective grid sizes being given in Table 7.1. The set of problems is chosen as in [5] and is designed so that when the number of processors increases the number of operations per processor in the LU factorization stays approximately constant when employing a nested dissection ordering [16].

TABLE 7.1
3D grid problems.

Processors	Rectangular grid sizes			Cubic grid size	Processors	Rectangular grid sizes			Cubic grid size
1	96	24	12	29	64	184	46	23	57
16	152	38	19	46	128	208	52	26	64
32	168	42	21	51	256	224	56	28	72
48	172	44	22	55	512	248	62	31	80

In Table 7.2, we show for the grid problems the distribution of work for type 1 masters (T1), type 2 masters (TM) and slaves (TS), and the type 3 root node (T3).

In particular for large problems, the work of the type 2 slaves becomes a major part of the overall work, and the candidate concept will have a great impact in those cases.

TABLE 7.2
Percentage distribution of work for 3D grid problems (nested dissection ordering).

	cubic								rectangular							
	LU				LDL^T				LU				LDL^T			
Procs.	T1	TM	TS	T3	T1	TM	TS	T3	T1	TM	TS	T3	T1	TM	TS	T3
1	100	0	0	0	100	0	0	0	100	0	0	0	100	0	0	0
16	18	5	63	14	18	2	65	15	25	5	58	12	25	2	61	12
32	7	4	75	14	8	1	77	14	16	4	68	12	16	2	70	12
48	7	4	75	15	8	1	77	14	14	4	70	12	12	2	74	12
64	5	3	78	14	5	1	81	13	10	4	74	12	10	1	77	12

7.2. General symmetric and unsymmetric matrices. The matrices described in this section all arise from industrial applications and include test matrices from the PARASOL Project [1], the Rutherford–Boeing Collection [12], and the University of Florida sparse matrix collection [9].

In Table 7.3, we describe the characteristics of the test matrices arising from real-life problems. We remark that in the case of the irregular problems, the work distribution heavily depends on the ordering used. The approximate minimum fill (AMF) ordering produces assembly trees that are rich in type 2 parallelism; on the other hand, the root nodes are so small that type 3 parallelism cannot be exploited effectively. On the other hand, METIS (as well as SCOTCH) provides more type 3 parallelism and, again, the major part of the work is associated with the factorization of type 2 nodes.

TABLE 7.3
Matrix order, type, and number of entries for the irregular test matrices.

Matrix name	Matrix type	Matrix order	No. of entries	Origin
audikw_1	symmetric	943695	39297771	PARASOL
bbmat	unsymmetric	38744	1771722	Rutherford-Boeing
bmwcra_1	symmetric	148770	10644002	PARASOL
ecl32	unsymmetric	51993	380415	University of Florida
g7jac200	unsymmetric	59310	837936	University of Florida
inline_1	symmetric	503712	18660027	PARASOL
pre2	unsymmetric	659033	5959282	University of Florida
ship003	symmetric	121728	8086034	PARASOL
twotone	unsymmetric	120750	1224224	University of Florida
xenon2	unsymmetric	157464	3866688	University of Florida

8. Experimental investigation of algorithmic details. In this section, we study the influence and scope of parameters in the algorithms used by Version 4.1 [4, 5] and by the new version of MUMPS. Furthermore, we present a detailed investigation of isolated parts of the improved algorithm by typical examples of phenomena that we have observed in our experiments.

8.1. The impact of $k_{\max}$ on communication and memory. We first show the impact of the parameter $k_{\max}$, defined in section 6.6, that controls the minimum granularity of the type 2 parallelism, on the volume of communication and memory.

We compare the behavior of Version 4.1 of MUMPS and the new code on one of the test matrices from section 7.1, corresponding to a cubic grid of order 46 and ordered by nested dissection. Here, we perform an LU factorization on an SGI Origin 2000 with 16 processors. This platform is well suited for testing the $k_{\max}$ parameter over a wide range of values due to its shared-memory architecture, where a large amount of memory is available to all processors.

The two graphs in the upper row of Figure 8.1 illustrate that with increasing $k_{\max}$, both the total volume of communication and the number of messages associated with dynamic scheduling decrease. This is due to the fact that a small $k_{\max}$ increases the required minimum number of slaves for a type 2 node, up to the point where the minimum granularity condition does not impact our scheduling.

The graph in the left lower corner of Figure 8.1 shows the increase in estimated and actually used memory with increasing $k_{\max}$, and the graph in the right lower corner shows the decomposition of the estimated memory into the space reserved for the communication buffers, the LU factors, and the stack. As potentially every processor can be selected as a slave during the factorization and the memory predicted depends monotonically on $k_{\max}$, the prediction during the analysis phase will lead to an increasing gap between real and estimated memory, as can be seen in the graph on the lower left. On the lower right, we see that the main contribution to the overestimation of the memory is the stack. As slaves stack their part of the contribution block until it can be received by the processors working on the parent of the node, the stack has to grow when $k_{\max}$ increases. Furthermore, a single type 2 slave is authorized to work on larger parts of a contribution block.

We now investigate the behavior of the new candidate-based code on the same test matrix where candidates are assigned without relaxation and layerwise redistribution, following the proportional mapping of the assembly tree.

From the two graphs in the bottom row of Figure 8.2 we observe the expected

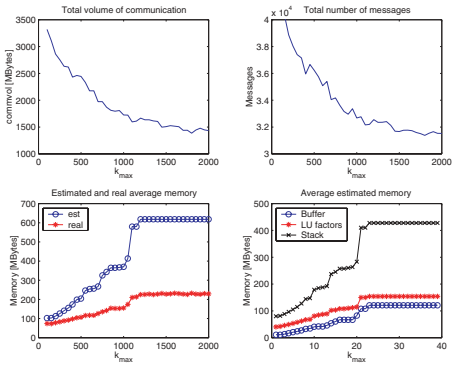


FIG. 8.1. *Impact of $k_{\max}$ on volume of communication and memory in Version* 4.1 *of MUMPS (Origin* 2000, 16 *processors).*

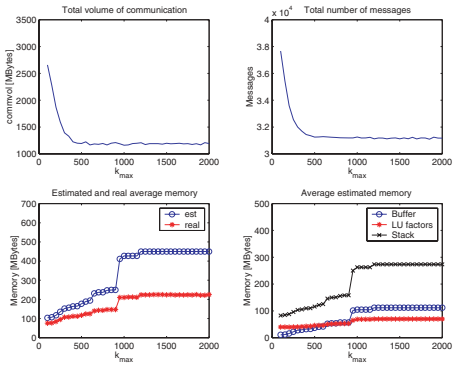


FIG. 8.2. *Impact of $k_{\max}$ on volume of communication and memory in the new version of MUMPS (Origin* 2000, 16 *processors).*

better estimation of memory. Furthermore, the two graphs in the top row of Figure 8.2 indicate that the communication volume in the new version of MUMPS drops faster with increasing $k_{\max}$ than it does for the previous version. This can be explained by the restricted freedom for the dynamic scheduling, so that actually less parallelism is created and fewer slaves are chosen during factorization. Thus, in the new code, we can choose a relatively small value of $k_{\max}$ and have the benefits of a relatively realistic memory estimation together with a reduced communication volume.

8.2. The impact of $k_{\max}$ on performance. In the following example, we show the impact of the parameter $k_{\max}$ on the factorization time.

Our test matrix comes from a cubic grid of order 51, ordered by nested dissection; we perform an *LU* factorization on a CRAY T3E with 64 processors. (Compared to Table 7.1, we have reduced the problem size to have enough flexibility with respect to memory for this parameter study.) Furthermore, because of limited memory and in order to separate the different algorithmic parameters, we use a candidate assignment without relaxation. The CRAY T3E is well suited for providing reliable timings for performance measures but has a distributed memory architecture with a fairly small amount of memory per processor.

From Figure 8.3, we see that with increasing $k_{\max}$ the factorization time decreases in both versions of the code as the dynamic scheduler has more freedom to decrease unnecessary parallelism. However, the previous version of MUMPS needs much more memory than the candidate-based version, and thus the flexibility for increasing $k_{\max}$ is more strictly limited. Once $k_{\max}$ is sufficiently large, a further increase in $k_{\max}$ shows no further improvements in performance. This corresponds to the results on the limited reduction in the volume of communication obtained in section 8.1.

8.3. Modifying the freedom offered to dynamic scheduling. We now investigate the behavior of the new code when modifying the assignment of candidates. We study two different approaches. As described in section 6.3, we can increase the number of candidates given to a node by relaxing its number of preferentials through the proportional mapping. Furthermore, according to Algorithm 6, we can modify the candidate assignment for a given layer by an optional redistribution of the candidates that takes account of the weight of the nodes relative to each other.

For our study, we use the same test case as in section 8.2. Figure 8.4 shows the

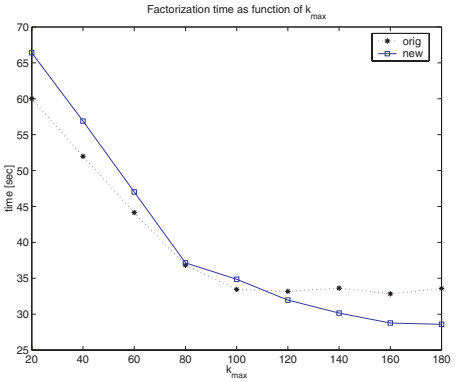


FIG. 8.3. *Impact of $k_{\max}$ on the performance of the LU factorization time for the original and the new version (CRAY T3E, 64 processors).*

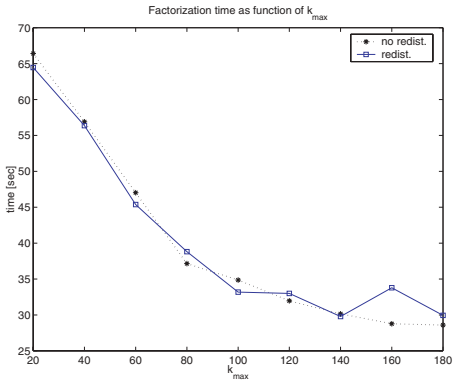


FIG. 8.4. *Comparison of the candidate assignment with (solid) and without (dotted) layerwise candidate redistribution when increasing minimum granularity (LU factorization time on CRAY T3E, 64 processors, no candidate relaxation).*

factorization time of the new version of MUMPS for the candidate assignment with and without layerwise candidate redistribution as a function of the minimum granularity. We cannot find significant differences in the behavior of the two approaches. This example is representative for the results we have obtained on the complete set of test problems.

We also have analyzed the impact of relaxation with and without layerwise redistribution on the volume of communication, memory, and performance. Our observation is that, with increasing relaxation, both the total volume of communication and the number of messages related to dynamic scheduling increase because the flexibility for choosing the slaves during factorization becomes greater. Likewise, the memory estimation grows with increasing relaxation. However, we do not observe a positive impact of relaxation on the performance of the algorithm; a possible interpretation is that, through the relaxation, we create additional parallelism that is not actually needed at run time. While this observation holds for all the experiments we have conducted, we are convinced that relaxation might show a positive impact on certain irregular problems from real-life applications. This has to be confirmed in future work.

8.4. Improved node splitting and amalgamation. In this section, we illustrate the additional capabilities for node splitting and node amalgamation and demonstrate the benefits. The former algorithm authorized node splitting only up to a fixed distance from the root node, where this distance depended only on the number of processors but not on the matrix. The new algorithm incorporates the splitting systematically in the upper part of the tree. We illustrate, in Table 8.1, the properties and benefits of the improved splitting in a selected case. The additional splitting slightly increases the number of assembly operations and the average amount of memory. However, it creates additional type 2 nodes. This significantly improves the performance and the memory balance.

Amalgamation in the previous version was possible only between a parent node and its oldest child; the greater freedom in the new code allows many more amalgamations. We illustrate in Table 8.2 the properties and benefits of the improved

TABLE 8.1
Comparison of the candidate-based LU factorization with and without improved node splitting (cubic grid of order 57 on CRAY T3E with 64 processors).

Algorithm	Nmb type 2	Operations (AMF) elim.	Operations (AMF) assem.	Mem. est. max	Mem. est. avg	Mem. real max	Mem. real avg	Facto. time
no splitting	126	7.98e+11	1.10e+09	196	143	141	92	182
with splitting	140	7.98e+11	1.30e+09	167	150	119	96	145

TABLE 8.2
Comparison of the candidate-based LDL^T factorization with and without improved node amalgamation (cubic grid of order 46 on CRAY T3E with 17 processors).

Type of amalgamation	Operations (ND) assem.	Operations (ND) elim.	Mem max	Mem avg	Mem real max	Mem real avg	Fact. time
Old	2.44e+08	5.91e+10	187	121	175	97	19.4
New	2.35e+08	5.91e+10	108	95	82	71	18.7

amalgamation to memory balance by a selected example. The additional amalgamation decreases the number of assembly operations and allows a better memory balance because the stacking of several large type 1 nodes can be avoided.

8.5. Postprocessing for a better memory balance. On the CRAY T3E, we show the benefits obtained by remapping the masters of type 2 nodes for better memory balance as described in section 6.5.

From Table 8.3, we see that the flop-based equilibration of the scheduling algorithm leads to severe memory imbalance both in the estimated and the actual memory. Performance reasons would encourage us to increase $k_{\max}$. However, this is impossible because of the strong memory imbalance. We observe that with postprocessing, the difference between average and maximum values for both the estimated and actual memory are much reduced. This allows us to double the $k_{\max}$ parameter for this test case and obtain better performance for the factorization.

TABLE 8.3
Memory (in megabytes) and factorization time (in seconds) of the candidate-based LU factorization with and without postprocessing (cubic grid of order 72 with nested dissection).

	No postprocessing					With postprocessing				
$k_{\max}$	Max est	Avg est	Max real	Avg real	Fact. time	Max est	Avg est	Max real	Avg real	Fact. time
80	179	117	172	102	165	136	117	123	102	152
160	Not enough memory					193	164	162	132	124

9. Performance analysis. In the following, we compare the performance of the new MUMPS code with the previous version [5] on the complete set of test problems presented in section 7.

9.1. Nested dissection ordering. In this section, we use the test matrices from Table 7.1 ordered by nested dissection. We have observed (see also results in Tables 9.1 and 9.3) that for up to 64 processors the new version has a similar performance to the good results obtained by the previous version. However, when more processors are used and the matrices become larger, the new code performs significantly better. Looking at the results on 128, 256, and 512 processors, we note the greatly improved scalability of the candidate-based code.

TABLE 9.1
Performance of the old and new LU factorization (time in seconds on the CRAY T3E).

	Cubic grids (ND)			Rect. grids (ND)		
Procs.	flops	old	new	flops	old	new
1	7.2e+09	23.2	23.2	4.5e+09	16.6	16.6
16	1.2e+11	30.8	31.8	7.3e+10	22.4	23.3
32	2.3e+11	43.3	42.2	1.4e+11	25.7	27.4
48	3.6e+11	53.0	57.5	1.8e+11	26.0	23.9
64	4.5e+11	59.0	52.9	2.4e+11	31.2	30.2
128	8.9e+11	93.4	72.7	4.9e+11	44.9	38.5
256	1.8e+12	163.5	119.4	7.7e+11	75.4	47.1
512	3.4e+12	599.6	189.1	1.4e+12	135.5	73.7

TABLE 9.2
Space for the LU factors (number of reals $\times 10^6$). The grid size and the number of processors as in Table 9.1.

	Cubic grids (ND)			Rect. grids (ND)		
	space	estim.	estim.	space	estim.	estim.
	used	old	new	used	old	new
	11.4	11.4	11.4	10.2	10.2	10.2
	77.9	84.3	78.6	70.4	82.3	70.9
	121.2	181.5	122.8	107.7	166.8	110.0
	165.9	289.5	170.7	130.2	255.5	134.7
	193.7	412.6	203.8	158.4	407.2	166.7
	309.7	897.9	357.0	260.1	1108.0	296.4
	504.4	2678.5	924.6	353.9	2420.5	478.0
	780.4	4594.0	1369.7	541.6	5759.0	921.5

Another major advantage of the new candidate-based code is that it better estimates the memory used for the factorization. In Table 9.2, we show the memory space for the LU factors of the old and the new versions of MUMPS. We see that the candidate-based code significantly reduces the overestimation of the storage required and that the gains increase with the matrix size and the number of processors.

The big gains of the new candidate-based code are a result of the individual improvements concerning splitting and amalgamation, reduced communication and the better locality of the computation as illustrated in section 8. Furthermore, we need to decrease $k_{\max}$ in the large problems for the old version of MUMPS because of memory. This limits the performance as we saw in section 8.2. On the other hand, we do not need to decrease $k_{\max}$ in the candidate-based code as the tighter estimates stay within the memory available.

As all regular test matrices are symmetric, we also can compare the old with the new candidate-based LDL^T factorization. The results presented in Table 9.3 confirm those obtained for the LU factorization. The candidate-based code shows a much better performance in particular for the large problems on a large number of processors due to improved locality of communication and computation and because of the bigger scope for increasing the $k_{\max}$ parameter.

TABLE 9.3
Performance of the LDL^T factorization (time in seconds on the CRAY T3E).

	Cubic grids (ND)			Rectangular grids (ND)		
Processors	flops	old	new	flops	old	new
1	3.6e+09	19.1	18.7	2.2e+09	13.5	13.1
16	5.9e+10	18.8	19.8	3.6e+10	13.8	13.2
32	1.1e+11	25.8	22.2	6.8e+10	15.5	15.3
48	1.8e+11	28.7	30.4	9.0e+10	14.2	14.8
64	2.2e+11	30.7	25.6	1.2e+11	17.6	16.8
128	4.4e+11	45.6	33.0	2.4e+11	33.5	20.3
256	9.1e+11	109.1	43.0	3.8e+11	45.2	18.4
512	1.7e+12	421.9	64.0	7.1e+11	195.5	24.3

Note that thanks to the improvements in the scalability of the new code, MUMPS now compares favorably to SuperLU on a large number of processors. The LU factorization times for SuperLU on 128 processors and the same nested dissection ordering, according to [5], are 71.1 seconds for the cubic and 56.1 seconds for the rectangular grid and should be compared to timings reported in Table 9.1, 72.7 seconds and 38.5

seconds, respectively.

9.2. AMF ordering. The AMF ordering [25, 30] produces trees that are difficult to exploit in MUMPS. The upper part of the tree, where type 2 and type 3 parallelism can be exploited, is usually a long and thin chain. As a typical example, we consider the case of an LU factorization on 64 processors. On cubic grids, the number of entries in the factors is 247.8×10^6 for AMF versus 193.8×10^6 for nested dissection and is thus larger. On rectangular grids, the number of entries in the factors is 148.1×10^6 for AMF and 158.4×10^6 for nested dissection. Here, AMF needs less space for the factors. However, the shape of the assembly tree still offers less potential for parallelism, and we expect the factorization time for AMF-ordered matrices to be considerably longer than for the case of nested dissection. This is confirmed by the results in Table 9.4, where we also show that the new strategy results in significant gains and that the absolute performance of the AMF ordering on rectangular grids is good on up to 128 processors.

TABLE 9.4
Performance of the LU and the LDL^T factorization (time in seconds on the CRAY T3E). $***$ indicates insufficient memory for analysis phase.

	LU factorization						LDL^T factorization					
	Cubic grids (AMF)			Rect. grids (AMF)			Cubic grids (AMF)			Rect. grids (AMF)		
Procs.	flops	old	new	flops	old	new	flops	old	new	flops	old	new
1	8.6e+09	25.7	25.7	3.1e+09	13.4	13.7	4.3e+09	19.5	19.5	1.6e+09	11.3	11.3
16	1.9e+11	55.5	54.9	5.4e+10	34.8	32.6	9.5e+10	29.0	29.2	2.7e+10	15.4	16.7
32	3.8e+11	96.3	81.3	1.0e+11	50.7	49.8	1.9e+11	34.1	33.8	5.1e+10	20.1	20.9
48	4.8e+11	114.6	98.2	1.9e+11	71.0	67.4	2.4e+11	36.3	36.5	9.3e+10	24.6	25.0
64	8.0e+11	188.0	145.4	1.8e+11	46.4	43.3	4.0e+11	51.8	48.6	8.9e+10	23.6	23.7
128	1.7e+12	302.6	242.7	4.6e+11	118.9	114.6	8.4e+11	86.1	67.8	2.3e+11	38.6	34.5
256	4.1e+12	740.9	484.1	8.6e+11	262.5	208.6	2.1e+12	237.7	117.3	4.3e+11	74.6	67.7
512	$***$			1.2e+12	325.7	264.7	$***$			6.2e+11	196.1	73.0

9.3. Performance analysis on general symmetric and unsymmetric matrices. In this section, we compare the performance of the new mapping algorithm with the previous version on general symmetric and unsymmetric matrices. The main problem with this comparison is that our algorithm offers the biggest performance gains only on a large number of processors. However, the unsymmetric matrices available to us are either too small to offer enough potential for scalability on more than 64 processors, or they are too large to do the analysis (which is performed on only one processor). This was already observed in the analysis of the scalability of both MUMPS and SuperLU [5] and is particularly important for the T3E architecture. However, while the problem in principle stays the same on the IBM SP3, the situation is alleviated due to the considerably larger amount of memory that is shared among the processors of an SMP node and thus available for the analysis phase.

For all orderings, we report the number of operations needed for the factorization phase in Table 9.5. From Table 9.6 we see that in general on the T3E, the new mapping algorithm performs similarly to the old one. As already noted, we would expect significant improvements on large matrices and on a number of processors greater than 64. However, we notice some improvements for the AMF ordering on `bbmat` and `g7jac200`. But since METIS generally provides better orderings, those improvements on AMF only show the capacity of our algorithm to correctly handle irregular trees.

TABLE 9.5
Number of operations for the factorization phases in Tables 9.6 *and* 9.7.

Matrix	AMF	METIS	Matrix	SCOTCH
bbmat	2.8e+10	2.8e+10	audikw_1	5.5e+12
bmwcra_1	9.9e+10	6.1e+10	bmwcra_1	6.4e+10
ecl32	3.5e+10	2.1e+10	g7jac200	1.6e+11
g7jac200	3.5e+10	5.5e+10	inline_1	1.4e+11
ship003	9.6e+10	8.3e+10	pre2	1.7e+11
twotone	2.9e+10	2.9e+10	ship003	9.3e+10
			xenon2	1.1e+11

TABLE 9.6
Performance of old and new code on the irregular test matrices (factorization time in seconds on the CRAY T3E).

Matrix	Order	Alg	8	16	32	64	Matrix	Order	Alg	8	16	32	64
bbmat	AMF	old	71.1	50.5	44.3	44.1	g7jac200	AMF	old	77.3	63.4	40.2	41.8
		new	69.6	44.1	27.6	21.7			new	78.3	61.3	38.6	33.7
	METIS	old	24.2	14.5	11.8	9.6		METIS	old	48.2	27.4	20.3	15.7
		new	22.2	14.1	10.8	8.8			new	41.4	26.7	19.9	13.6
bmwcra_1	AMF	old	-	44.6	30.3	27.6	ship003	AMF	old	66.0	34.0	24.4	22.1
		new	-	42.4	28.5	26.9			new	62.2	33.5	24.2	20.4
	METIS	old	36.6	20.1	13.5	8.5		METIS	old	-	29.2	18.2	12.3
		new	35.7	20.9	13.2	8.4			new	-	28.4	18.0	12.0
ecl32	AMF	old	25.7	19.9	16.6	16.0	twotone	AMF	old	47.1	28.3	20.8	19.1
		new	24.2	19.0	16.0	14.5			new	47.4	29.0	20.9	18.7
	METIS	old	16.7	10.7	7.7	6.3		METIS	old	26.9	19.1	13.3	11.4
		new	16.0	11.4	7.7	5.6			new	27.9	17.7	11.9	11.2

TABLE 9.7
Performance of old and new code on large irregular test matrices (factorization time in seconds on the IBM SP3).

Matrix	Order	Alg	16	32	64	128
audikw_1	SCOTCH	old	-	-	300.9	211.1
		new	-	-	289.7	192.4
bmwcra_1	SCOTCH	old	8.8	7.6	8.3	7.7
		new	9.5	7.9	6.8	4.5
g7jac200	SCOTCH	old	28.0	37.2	36.6	39.7
		new	29.4	26.4	30.7	28.1
inline_1	SCOTCH	old	20.7	14.6	12.8	12.6
		new	17.6	14.0	9.3	7.4
pre2	SCOTCH	old	35.7	31.6	29.7	31.6
		new	31.7	27.3	24.7	26.4
ship003	SCOTCH	old	13.7	13.9	11.2	12.4
		new	13.6	11.1	7.6	7.1
xenon2	SCOTCH	old	16.2	15.6	10.7	12.5
		new	16.5	13.0	10.5	10.6

TABLE 9.8
*Time for mapping and complete analysis phase for large irregular test matrices (*64 *processors, time in seconds on the IBM SP3).*

Matrix	Order	Alg	Map.	Anal.
audikw_1	SCOTCH	old	0.6	139.7
		new	2.2	147.0
bmwcra_1	SCOTCH	old	0.1	15.3
		new	0.3	15.7
g7jac200	SCOTCH	old	0.1	7.1
		new	0.2	7.2
inline_1	SCOTCH	old	0.3	62.2
		new	0.9	63.6
pre2	SCOTCH	old	1.2	133.3
		new	4.6	135.8
ship003	SCOTCH	old	0.1	5.9
		new	0.3	6.6
xenon2	SCOTCH	old	0.1	10.0
		new	0.3	10.5

In Table 9.7, we show the performance results for the largest irregular test matrices on the IBM SP3. Here, SCOTCH proved to be the most suitable ordering. Furthermore, we compare the costs for the mapping in terms of computing time relative to the costs for the complete analysis phase in Table 9.8.

We see that on the IBM SP3 we significantly improve the scalability on the large problems that we were not able to run on the CRAY T3E. Comparing the time needed

for the tree mapping in the analysis phase as reported in Table 9.8, we note that the new mapping algorithm is approximately three times as expensive as the old one. Given that the absolute cost for the mapping is small, the benefits from the improved factorization fully justify the new approach.

10. Perspectives and future work. In this section, we summarize the open questions that need further investigation.

In section 8.3, we investigated the behavior of the new code when modifying the assignment of candidates through relaxation and layerwise redistribution. On the test cases that we have studied in the framework of this paper, these modifications have not shown a significant positive effect on the overall performance of the code. Still, there is an intuitive argument suggesting further experiments. The analysis phase tries to predict the actual factorization of the matrix and takes mapping decisions based on this symbolic factorization. However, there are cases where this approach might not be accurate enough; for example, we do not take into account costs of communication between the processors as is done, for example, by the static scheduler of PaStiX [19]. A correction of the mapping decisions that combines the techniques presented in this paper could result from the following observation. Since, during factorization, the assembly tree is treated from bottom up, we might expect mapping problems to have more severe influence towards the root of the tree. For this reason, we could decide to offer more freedom to dynamic scheduling near the root nodes so that unfortunate mapping decisions can be corrected dynamically there.

Finally, our candidate-based approach can be modified to take account of the system architecture, for example, with respect to nonuniform communication costs on machines consisting of SMP nodes. We can modify the task scheduling so that processors which require expensive communications are penalized so that the master-slave communication costs are reduced. This approach is further described in [6].

11. Summary and conclusions. Previous studies of MUMPS, a distributed memory direct multifrontal solver for sparse linear systems, indicated that its scalability with respect to computation time and use of memory should be improved. In this paper, we have presented a new task scheduling algorithm designed to address these problems. It consists of an approach that treats the assembly tree layer by layer and integrates tree modifications, such as amalgamation and splitting, with the mapping decisions. As a major feature, we have adapted the concept of candidate processors that are determined during the analysis phase of the solver in order to guide the dynamic scheduling during the factorization.

We have illustrated key properties of the new algorithm by detailed case studies on selected problems. Afterward, by comparison of the old code with the new code on a large set of regular and irregular test problems, we have illustrated the main benefits of the new approach. These include an improved scalability on a large number of processors, reduced memory demands and a smaller volume of communication, and the easier handling of parameters relevant for the performance of the algorithm. Finally, we have pointed out possible extensions of our algorithm, in particular with respect to its use on SMP architectures.

Acknowledgments. We are grateful to E. Ng for providing access to the CRAY T3E at NERSC. J. Koster and J. Y. L'Excellent gave helpful comments on an earlier version of this paper. S. Pralet provided additional test results on the IBM SP3 from CINES, Montpellier, France. The referees gave detailed and helpful remarks on how to improve the presentation and the contents of this paper.

REFERENCES

[1] *PARASOL test data*, http://www.parallab.uib.no/parasol/data.html (last updated May 23, 2003).

[2] P. R. Amestoy and I. S. Duff, *Memory management issues in sparse multifrontal methods on multiprocessors*, Int. J. Supercomput. Appl., 7 (1993), pp. 64–82.

[3] P. R. Amestoy, I. S. Duff, and J. Y. L'Excellent, *Multifrontal parallel distributed symmetric and unsymmetric solvers*, Comput. Methods in Appl. Mech. Engrg., 184 (2000), pp. 501–520.

[4] 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.

[5] P. R. Amestoy, I. S. Duff, J. Y. L'Excellent, and X. S. Li, *Analysis, tuning and comparison of two general sparse solvers for distributed memory computers*, ACM Trans. Math. Software, 27 (2001), pp. 388–421.

[6] P. R. Amestoy, I. S. Duff, S. Pralet, and C. Vömel, *Adapting a parallel sparse direct solver to architectures with clusters of SMPs*, Parallel Comput., 29 (2003), pp. 1645–1668.

[7] C. Ashcraft and R. G. Grimes, *The influence of relaxed supernode partitions on the multifrontal method*, ACM Trans. Math. Software, 15 (1989), pp. 291–309.

[8] J. Choi, J. Demmel, I. Dhillon, J. Dongarra, S. Ostrouchov, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley, *ScaLAPACK: A portable linear algebra library for distributed memory computers: Design issues and performance*, Comput. Phys. Comm., 97 (1996), pp. 1–15.

[9] T. A. Davis, *University of Florida Sparse Matrix Collection*, http://www.cise.ufl.edu/research/sparse/matrices (last updated August, 2004). NA Digest, 92 (42), October 16, 1994. NA Digest, 96 (28), July 23, 1996. NA Digest, 97 (23), June 7, 1997.

[10] 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.

[11] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Oxford University Press, London, 1986.

[12] I. S. Duff, R. G. Grimes, and J. G. Lewis, *The Rutherford-Boeing Sparse Matrix Collection*, Technical report RAL-TR-97-031, Atlas Centre, Rutherford Appleton Laboratory, 1997. Also Technical report ISSTECH-97-017 from Boeing Information & Support Services and Report TR/PA/97/36 from CERFACS, Toulouse, France, http://www.cerfacs.fr/algor/reports/index.html.

[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, *The multifrontal solution of unsymmetric sets of linear equations*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 633–641.

[15] A. Geist and E. Ng, *Task scheduling for parallel sparse Cholesky factorization*, Int. J. Parallel Program., 18 (1989), pp. 291–314.

[16] A. George and E. Ng, *SPARSPAK: Waterloo Sparse Matrix Package User's Guide for SPARSPAK-B*, Research report CS-84-37, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 1984.

[17] J. A. George, J. W. H. Liu, and E. G.-Y. Ng, *Communication results for parallel sparse Cholesky factorization on a hypercube*, Parallel Comput., 10 (1989), pp. 287–298.

[18] A. Gupta, *Recent advances in direct methods for solving unsymmetric sparse systems of linear equations*, ACM Trans. Math. Software, 28 (2002), pp. 301–324.

[19] P. Hénon, P. Ramet, and J. Roman, *PaStiX: A high-performance parallel direct solver for sparse symmetric definite systems*, Parallel Comput., 28 (2002), pp. 301–321.

[20] L. A. Hall, *Approximation algorithms for scheduling*, in Approximation Algorithms for NP-Hard Problems, D. Hochbaum, ed., PWS Publishing, Boston, 1996, pp. 1–45.

[21] G. Karypis and V. Kumar, *MeTis: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, Version* 4.0, University of Minnesota, Minneapolis, 1998.

[22] J. W. H. Liu, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 134–172.

[23] J. W. H. Liu, *The multifrontal method for sparse matrix solution: Theory and practice*, SIAM Rev., 34 (1992), pp. 82–109.

[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] 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.
[26] F. Pellegrini and J. Roman, *Sparse matrix ordering with Scotch*, in Proceedings of High Performance Computing and Networking 1997, Vienna, Austria, Lecture Notes in Comput. Sci. 1225, Springer-Verlag, New York, 1997, pp. 370–378.
[27] F. Pellegrini, J. Roman, and P. Amestoy, *Hybridizing nested dissection and halo approximate minimum degree for efficient sparse matrix ordering*, Concurrency: Practice and Experience, 12 (2000), pp. 69–84.
[28] A. Pothen and C. Sun, *A mapping algorithm for parallel sparse Cholesky factorization*, SIAM J. Sci. Comput., 14 (1993), pp. 1253–1257.
[29] P. Raghavan, *Distributed Sparse Matrix Factorization: QR and Cholesky Decompositions*, Ph.D. thesis, Pennsylvania State University, University Park, PA, 1991.
[30] 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. 26, No. 2, pp. 566–582

A KRYLOV SUBSPACE METHOD FOR INFORMATION RETRIEVAL*

KATARINA BLOM† AND AXEL RUHE‡

Abstract. A new algorithm for information retrieval is described. It is a vector space method with automatic query expansion. The original user query is projected onto a Krylov subspace generated by the query and the term-document matrix. Each dimension of the Krylov space is generated by a simple vector space search, using first the user query and then new queries generated by the algorithm and orthogonal to the previous query vectors.

The new algorithm is closely related to latent semantic indexing (LSI), but it is a local algorithm that works on a new subspace of very low dimension for each query. This makes it faster and more flexible than LSI. No preliminary computation of the singular value decomposition (SVD) is needed, and changes in the data base cause no complication.

Numerical tests on both small (Cranfield) and larger (*Financial Times* data from the TREC collection) data sets are reported. The new algorithm gives better precision at given recall levels than simple vector space and LSI in those cases that have been compared.

Key words. information retrieval, vector space model, query expansion, latent semantic indexing, singular value decomposition, Lanczos algorithm, Krylov subspace

AMS subject classifications. 68P20, 65F15, 65F20

DOI. 10.1137/S0895479803392261

1. Introduction.

The purpose of an information retrieval (IR) system is to seek through a large collection of information items, or *documents*, to retrieve those relevant to information requests, or *queries*, stated by a user. In the present contribution, we will show how computational tools from numerical linear algebra can be helpful. We will use IR criteria to decide success or failure of the algorithms developed: What proportion of the relevant documents are found, and how many of the retrieved documents are relevant to the user?

The documents may be books in a library, entries in a data base of news telegrams, scientific papers in journals, or web pages on the World Wide Web (WWW). Each document contains *terms*, words that are significant in some way. The query is also formulated in terms of the same kind. We will look at the document collection as a huge matrix, where there is one row for each term that occurs anywhere in the collection and each column represents one document. This *term-document* matrix is denoted A throughout this paper. We let the element a_{ij} in row i and column j of A be nonzero if the ith term is present in document number j, and zero otherwise. The term-document matrix will typically be very large and very sparse. The query will be expressed in the same terms as the documents, i.e., as a column vector q, where the ith element q_i is nonzero if the ith term is a part of the query, and zero otherwise.

A very simple IR algorithm is to choose those documents that contain any of the terms in the query. This *Boolean search* can be expressed as a row vector $p^T = q^T A$, where each element p_j is the scalar product between the query vector q and a document

*Received by the editors December 15, 2003; accepted for publication (in revised form) by D. Boley April 27, 2004; published electronically January 12, 2005.
http://www.siam.org/journals/simax/26-2/39226.html

†Department of Computing Science, Chalmers Institute of Technology and the University of Göteborg, SE-41296 Göteborg, Sweden (blom@cs.chalmers.se). The research of this author was supported by The Swedish Research Council, Vetenskapsrådet, contract 2002-4152.

‡Department of Numerical Analysis and Computer Science, NADA, Royal Institute of Technology, SE-10044 Stockholm, Sweden (ruhe@kth.se).

column vector a_j of A, and choosing those documents for which p_j is nonzero. (We use the common linear algebra convention of letting a Latin letter stand for a column vector and ^{T} stand for transposing a column into a row. The matrix A has the columns $A = [a_1, a_2, \ldots, a_n]$.)

The *vector space model* is a refinement of Boolean search. The numerical values of the scalar products p_j are used to get angles between the query vector q and the document vectors a_j. The documents are scored, starting with those that make the smallest angle to the query vector.

In the present contribution we will study refinements of the vector space model. The main emphasis is on *subspace methods*, where we project the query and document vectors on a carefully chosen subspace and use the angles between these projected vectors to determine closeness. We show that in many cases subspace methods behave similarly to methods based on *query expansion*, another common class of refined vector space methods.

One subspace method is latent semantic indexing (LSI) [8], where the dominant principal component subspace computed by the singular value decomposition (SVD) is used. It is supposed to filter away noisy and particular information from the general and relevant information that we need to distinguish between documents on different subjects. Another subspace method is based on a known classification and uses *concept vectors* [6, 13]. One may also apply a probability model; this leads to computing convex combinations of nonnegative basis vectors [12, 2].

The purpose of this contribution is to develop a new subspace method based on *Krylov sequences* of subspaces reachable from the query vector. The first steps of the Krylov sequence correspond to a query expansion that is closely related to query expansion based on co-occurrences as introduced by Sparck Jones [14] and studied by Xu and Croft [15].

The advantages of our approach, as compared to LSI, are that it works on the original term-document matrix A, no SVD computation is needed in the outset, and it is trivial to add and delete terms and documents between queries. The main computational work is the same as a few applications of a naive vector space search; the rest is manipulation of small matrices.

1.1. Summary of contents. After some preliminary explanations of numerical linear algebra and IR notation in this section, we describe subspace methods in section 2. We explain their common characteristics and show that some well-known algorithms can be characterized as subspace methods, using different subspaces. We also discuss the relation between subspace methods and query expansion. In section 3 we describe the Krylov subspace algorithm we have used. It is simply the well-known Golub–Kahan bidiagonalization [9] applied to the term-document matrix A, starting at the query q. It is used to find an expanded query $\hat{q}$, which is used to compute angles for scoring the document vectors a_j. We also give quantities that can be used to determine convergence. In our context the algorithm is stopped at a much earlier stage than, for instance, when solving least squares problems. Finally, in section 4, we show results of some numerical experiments, using both the small and well-known Cranfield data and a larger test matrix coming from the *Financial Times* collection in the TREC material (from a text retrieval conference) [11].

We have formulated our algorithm and gotten some preliminary results in the licentiate thesis of the first author [3]. Further developments, like term weighting, experiments on more data sets, and the inclusion of relevance feedback, are discussed in the thesis [4]. Experiments on small matrices are reported in more detail in the conference contribution [5].

1.2. Notation.

Matrices. Throughout this paper, A will denote the $m \times n$ term-document matrix. The jth column vector of the matrix A will be denoted a_j, and the jth column vector of the identity matrix I will be denoted e_j.

Singular value decomposition. Let

$$A = U\Sigma V^T \tag{1.1}$$

be the SVD of A; see [10]. The best rank s approximation to A in the Frobenius or sum of squares norm is

$$A^{(s)} = U_s \Sigma_{ss} V_s^T, \tag{1.2}$$

where U_s and V_s are formed by the first s columns of U and V and the $s \times s$ diagonal matrix Σ_{ss} has the s largest singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_s$ in its diagonal.

Seen as a mapping, the $m \times n$ matrix A maps the n-dimensional space $\mathcal{R}^n$ into its range space $\mathcal{R}(A)$, the subspace of $\mathcal{R}^m$ which is spanned by the columns of A. Its dimension is r, the rank of A.

Krylov spaces. A Krylov subspace of a square matrix C, starting at the vector v, is a subspace of the form

$$\mathcal{K}_r(C, v) = \text{span}\{v, Cv, C^2v, \ldots, C^{r-1}v\}. \tag{1.3}$$

Increasing the dimension r, we finally get the entire reachable subspace of the pair (C, v). Its dimension is $r \leq n$, the dimension of v.

Measures. Two standard measures used by the IR community are *precision* and *recall*. Precision is the ratio of the number of relevant documents retrieved for a given query over the total number of documents retrieved. Recall is the ratio of relevant documents retrieved over the total number of relevant documents for that query. Precision and recall are usually inversely related (when precision goes up, recall goes down and vice versa). A recall level for a particular query can be arbitrarily chosen from $\frac{1}{t}, \frac{2}{t}, \ldots, 1$, where t is the number of documents relevant to this particular query.

In order to show precision at various recall levels graphically, interpolation may be used. The *interpolated precision* at a recall cutoff R for one query is defined to be the maximum precision at all recall levels greater than or equal to R.

The *average precision* is a single valued measure that reflects performance over all relevant documents. Average precision is the average of the precision value obtained after each relevant document is retrieved. Average precision will reward systems that rank all relevant documents high; the last relevant document found is equally important as the first.

When reporting results for test sets with multiple queries, we will consider the *mean interpolated average precision* over all queries at a fixed sequence of recall cutoff values.

A way to compare performance when finding the first relevant documents is *document level average*, DLA(i), the precision when a certain number, i, of documents are retrieved. It mimics the use of a search engine where 10 documents are presented to the user each time. Then DLA(10) is the fraction of those that are relevant. For further details, see Harman [11].

Relevance is always judged by comparing the results of an algorithm to relevance judgments provided with the test sets. These have been compiled by a panel of human experts who have considered at least all those documents marked as relevant.

2. Subspace methods. In a general sense, the vector space method works in a space $\mathcal{D}$ of all documents that can be expressible as texts. This space of all possible documents has a countably infinite number of dimensions, and it is not simple to determine closeness between two documents. We therefore choose to see each document as a bag of terms, and represent it as a vector $a_j \in R^m$ in the m-dimensional space of document vectors. This is already a rather severe restriction: we have reduced the dimension from infinity to m. We have also made a choice of which words we regard as significant, and used these words as terms.

When terms are chosen, we represent the query as a vector $q \in R^m$. We use angles between the query vector q and the document vectors a_j to determine which documents to retrieve in the naive vector space method.

In our IR task, we have a finite collection of n documents to choose from; they build up a document collection space $\mathcal{A} = \mathcal{R}(A)$, the range space of the term-document matrix A, which is of dimension at most n. Most often the number of terms m is larger than the number of documents, $m > n$, and the documents are linearly independent, making $\mathcal{A}$ into an n-dimensional subspace $\mathcal{A} \subset R^m$. The query vector q is not in this subspace $\mathcal{A}$, but we may use the projected query vector $P_{\mathcal{A}}q$ and retrieve those documents a_j that are closest to that vector. If we use angles in the Euclidean space to decide closeness, this will yield the same ranking as when we use the angles between the document vectors and the original query vector.

A wide class of IR algorithms can now be classified as *subspace algorithms* where we restrict our view to a subspace $\mathcal{S} \subset \mathcal{A}$ and use angles between a projected query $\hat{q} = P_{\mathcal{S}}q$ and projected documents $\hat{a}_j = P_{\mathcal{S}}a_j$.

Let us look at some natural choices of subspaces $\mathcal{S}$ in what follows.

2.1. Dominant subspace: LSI. LSI [8] uses the SVD (1.1) of the term-document matrix

$$A = U\Sigma V^T$$

and chooses the space of the leading s singular vectors (1.2),

$$\mathcal{S} = \text{span}\,[U_s]\,.$$

It separates the global and general structure, corresponding to the large singular vectors, from local or noisy information, which hides among the small. LSI has been reported to perform quite well on both rather large and small document collections. See, for example, Dumais [7]. It can handle synonymy (when two words mean the same thing) and polysemy (when one word has several distinct meanings depending on context) quite well. However, LSI needs substantial computational work to get the SVD, and there is no simple way to determine how many singular vectors s are needed to span the leading subspace. Work on this has been done by Berry [1] and by Zha, Marques, and Simon [16].

2.2. Classification: Centroid vectors. The singular vectors make up a basis of the best rank s approximation to the given term-document matrix A, and this can be considered as the best subspace if nothing else is known. On the other hand, if we know that the documents are taken from a set of subclasses, we may use a carefully selected set of centroid or *concept vectors* as the basis of another subspace $\mathcal{S}$; see Dhillon and Modha [6]. Park, Jean, and Rosen [13] compare the use of singular and centroid vectors in a general formulation of low rank approximations of the term-document matrix A.

2.3. Reachable subspaces: Krylov sequences. In the present contribution, we will try a third sequence of subspaces. We will let the subspaces be determined by the query vector q, taking the Krylov sequence of subspaces of vectors reached from q via a small number k of naive vector space searches.

In matrix language, this means that we take the query vector q, and multiply it with the transposed term-document matrix A to get a ranking or *scoring vector* $p = A^T q$. Each element p_j of p is a scalar product between the query vector q and the corresponding document vector a_j, so the elements of p give a ranking from the naive vector space method (if the columns of A are normalized). In this first step of the Krylov sequence, we find those documents that are directly related to the query, let us say its *sisters*.

In the second step, we multiply this scoring vector p with the term-document matrix A to get a new vector $q_2 = Ap$, a new query that contains all the terms that were contained in the documents that p pointed to. If we apply this new query, we get $p_2 = A^T q_2$, which points to all documents that contain any of all the terms in q_2, i.e., those two links away from the query, let us say its *cousins*.

In later steps this continues in a chain letter fashion, and soon we will reach all documents in the collection that are *reachable* from the query, to borrow a term from control theory. In matrix language,

$$\mathcal{S} = \mathcal{K}_k(AA^T, q) \tag{2.1}$$

after k steps; see (1.3).

In our computation we not only follow the Krylov sequence, but also make the vectors $q_1, q_2, \ldots, q_r$ and $p_1, p_2, \ldots, p_r$ into *orthogonal* bases. Intuitively this means that we remember what we asked for in the first query, q_1, and make a totally different query next time, q_2. This is standard practice in numerical linear algebra.

2.4. Relevant subspaces. There is a fourth subspace that is of theoretical interest and can be used for comparison purposes. That is the *relevant subspace* $\mathcal{Z}$ spanned by those documents that are relevant to the query q. This subspace is not possible to use in any practical algorithm; it supposes that all the relevant documents are already known. However, it is interesting to see whether the query q is closer to the relevant subspace $\mathcal{Z}$ than to any other subspace spanned by a similar number of document vectors. Are there many irrelevant documents that are closer to the relevant subspace $\mathcal{Z}$ than the query q?

In a way, the properties of the relevant subspace determine whether there is any hope for any algorithm, built up by tools from numerical linear algebra, to find the documents relevant to a given query.

2.5. Subspaces and query expansion. Subspace algorithms are closely related to another class of refined vector space IR methods built up around *query expansion*. Say that the subspace algorithm takes a subspace $\mathcal{S}$ in any of the manners described in the previous subsections, and uses the angles between the projected query $\hat{q} = P_{\mathcal{S}} q$ and the projected documents $\hat{a}_j = P_{\mathcal{S}} a_j$ to determine which documents a_j are relevant to the query q. The cosine of this angle is

$$\hat{c}_j = \frac{\hat{q}^T \hat{a}_j}{\|\hat{q}\|_2 \|\hat{a}_j\|_2}.$$

The scalar product in the numerator is

$$\hat{q}^T \hat{a}_j = (P_{\mathcal{S}} q)^T P_{\mathcal{S}} a_j = q^T P_{\mathcal{S}}^T P_{\mathcal{S}} a_j = q^T P_{\mathcal{S}} a_j = (P_{\mathcal{S}} q)^T a_j = \hat{q}^T a_j,$$

provided that the projection is orthogonal, $P^T = P$. We see that the scalar product between the projected query vector $\hat{q}$ and the projected document vector $\hat{a}_j$ is the same as that between the projected query $\hat{q}$ and the *original* document vector a_j. Using scalar products to determine closeness, the subspace method based on $\mathcal{S}$ gives the same result as a straightforward vector space method using the expanded query $\hat{q}$. The angles are not invariant, however, since the norms in the denominator differ. We know that $\|\hat{a}_j\|_2 \leq \|a_j\|$, giving a larger cosine or smaller angle in the subspace than in the query expansion case.

Still, the result of a subspace method based on $\mathcal{S}$ is closely related to using the expanded query $\hat{q} = P_{\mathcal{S}} q$ in the original vector space method.

When we choose $\mathcal{S}$ as a Krylov subspace (2.1), our choice of query expansion is related to the technique of Sparck Jones [14]. The second vector in the Krylov sequence (2.1), $\tilde{q}_2 = AA^T q$, weighs in components of all terms that are co-occurring with the terms in the original query. The weights give an emphasis to the co-occurrence in the documents that are ranked highest in the vector space search, $p = A^T q$, giving an effect similar to the local expansions of Xu and Croft [15].

3. The Krylov subspace algorithm. We use the Golub–Kahan bidiagonalization algorithm [9] to compute the Krylov sequence of subspaces (2.1). It is a variant of the Lanczos tridiagonalization algorithm and is widely used in the numerical linear algebra community.

The Golub–Kahan algorithm starts with the normalized query vector $q_1 = q/\|q\|$ and computes two orthonormal bases P and Q, adding one column for each step k; see [10, section 9.3.3].

ALGORITHM BIDIAG.
Start with $q_1 = q/\|q\|_2\,, \beta_1 = 0$.
For $k = 1, 2, \ldots, r$ *do*
1. $\alpha_k p_k = A^T q_k - \beta_k p_{k-1}$
2. $\beta_{k+1} q_{k+1} = A p_k - \alpha_k q_k$

End.

The scalars α_k and β_k are chosen to normalize the corresponding vectors.

Define

$$
\begin{aligned}
Q_{r+1} &= \begin{bmatrix} q_1 & q_2 & \ldots & q_{r+1} \end{bmatrix}, \\
P_r &= \begin{bmatrix} p_1 & p_2 & \ldots & p_r \end{bmatrix}, \\
B_{r+1,r} &= \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & & \alpha_r \\ & & & \beta_{r+1} \end{bmatrix}.
\end{aligned} \tag{3.1}
$$

After r steps we have the *basic recursion*,

$$
\begin{aligned}
A^T Q_r &= P_r B_{r,r}^T, \\
A P_r &= Q_{r+1} B_{r+1,r}.
\end{aligned}
$$

The columns of Q_r will be an orthonormal basis of the Krylov subspace (2.1),

$$
\operatorname{span}[Q_r] = \mathcal{K}_r(AA^T, q) \subseteq \mathcal{R}([A\, q]), \tag{3.2}
$$

in the document space, spanned by the query q and the columns of A. The columns of P_r similarly span a basis of the Krylov subspace,

$$\text{span}\,[P_r] = \mathcal{K}_r(A^TA, A^Tq) \subseteq \mathcal{R}(A^T), \tag{3.3}$$

in the term space spanned by the rows of A.

We see that $B_{r+1,r} = Q_{r+1}^T AP_r$ is the projection of A into these Krylov subspaces, and the singular values of $B_{r+1,r}$ will be approximations to those of A.

If $\beta_k = 0$ for some $k \leq r$, we have exhausted the Krylov space (3.2), reachable from the query q. Then $Q_k B_{k,k} P_k^T$ is the restriction of A to this reachable subspace, and the singular values of $B_{k,k}$ are a subset of those of A.

The columns of AP_r span the *reached subspace* after r steps starting from q. It is the intersection between the Krylov subspace (3.2) and the column space of A,

$$\mathcal{R}(AP_r) = \text{span}\,[Q_{r+1}B_{r+1,r}] \subseteq \mathcal{R}(A). \tag{3.4}$$

The basic recursion (3.2) implies that the rescaled subspace has the orthonormal basis W_r, where

$$W_r = Q_{r+1}H_{r+1,r}\,, \tag{3.5}$$

with $H_{r+1,r+1}$ the orthogonal factor in the QR factorization,

$$B_{r+1,r} = H_{r+1,r+1}R\,. \tag{3.6}$$

Note that since $B_{r+1,r}$ is bidiagonal, $H_{r+1,r+1}$ will be both orthogonal and Hessenberg and can be computed as a product of r elementary rotations.

The projected query vector. It is now easy to use the basis W_r (3.5) to project the query and the documents into the reached subspace (3.4). The projected query $\hat{q}$ is

$$\hat{q} = P_{\mathcal{R}(AP_r)}q = W_rW_r^Tq = W_rH_{r+1,r}^Te_1 = W_r\begin{pmatrix} h_{1,1} \\ h_{1,2} \\ \vdots \\ h_{1,r} \end{pmatrix}, \tag{3.7}$$

and we see that the first row of H gives the coordinates of the query in the basis W. When we run several steps r of our algorithm, new columns are added to H, but when one column $r+1$ is added in step r, it is only the last rth column that is modified.

We get the projected document $\hat{a}_j$ similarly as

$$\hat{a}_j = W_rW_r^Ta_j. \tag{3.8}$$

3.1. Scoring documents. We may regard our algorithm as a subspace method and choose the angles between the query and each of the document vectors, projected onto the reached subspace (3.4),

$$\text{Css}_j^{(r)} = \frac{\hat{q}^T\hat{a}_j}{\|\hat{q}\|_2\|\hat{a}_j\|_2}, \qquad j = 1, \ldots, n. \tag{3.9}$$

Alternatively, we may regard our algorithm as a query expansion method and use the angles between the projected query and the original documents,

$$\text{Cqe}_j^{(r)} = \frac{\hat{q}^Ta_j}{\|\hat{q}\|_2\|a_j\|_2}, \qquad j = 1, \ldots, n. \tag{3.10}$$

We compute these quantities using the basis W from (3.5) and the small orthogonal Hessenberg $H_{r+1,r+1}$ of (3.6). Apply an elementary orthogonal transformation S_r to make all elements but the first in the first row of $H_{r+1,r}S_r$ zero. Then W_rS_r forms a new basis of the reached subspace (3.4). The first element $(y_j^{(r)})_1$ in the vector

$$y_j^{(r)} = S_r^T W_r^T a_j$$

will give the component of a_j along $\hat{q}$ and the rest of the projected $\hat{a}_j$ (3.8) as the norm of the remaining elements in $y_j^{(r)}$. Thus the subspace cosine (3.9) is

$$\mathrm{Css}_j^{(r)} = \frac{(y_j^{(r)})_1}{\|y_j^{(r)})\|_2},$$

while the query expansion cosine (3.10) is slightly smaller at

$$\mathrm{Cqe}_j^{(r)} = \frac{(y_j^{(r)})_1}{\|a_j\|_2}.$$

Our experiments have shown that using the query expansion cosines Cqe_j (3.10) of the angles between projected query and original documents for scoring often gives better performance than the subspace cosines Css_j (3.9), so we use query expansion, Cqe_j, as our standard. It gives a preference for documents whose vectors a_j are closer in angle to the reached subspace.

3.2. Following progress. In the Krylov method, a new bidiagonalization is performed for every query vector q. Thus the number of iterations must be small. The optimal number of iterations r is different for various queries. Choosing the optimal number r of iterations is an interesting and important problem. Figure 3.1 shows performance for the Cranfield set using different numbers of iterations r. Performance is measured by average precision. It is clear from this figure that best average performance for all queries is reached when three iterations are performed. When more than three iterations are used, the performance rapidly converges towards the performance of the vector model. Note that some queries show optimal performance after two iterations and very few after one iteration. For one iteration, performance is worse than the performance for the vector model for most queries. This pattern of performance (initial worse than the vector model, increasing performance, and then a rapid convergence towards the vector model) was observed for most of the queries in all data sets we tested.

The convergence towards the vector model performance can easily be explained and estimated using quantities from the bidiagonalization algorithm presented.

Consider the least squares problem

$$\min_x \|Ax - q\|_2, \tag{3.11}$$

where A is the term-document matrix and q is the query vector. It can be solved using the BIDIAG algorithm (see, for example, the textbook [10]). In step k the distance between the query vector and the projected query vector $\hat{q}^{(k)}$ is the residual

$$d^{(k)} = q - Ax^{(k)} = q - \hat{q}^{(k)}.$$

Here $x^{(k)}$ is the solution to problem (3.11) in step k. The distance decreases as we let k grow, but will not tend to zero unless the query is a linear combination of the documents in A.[1]

[1] In our tests no query vector is completely in the range of A.

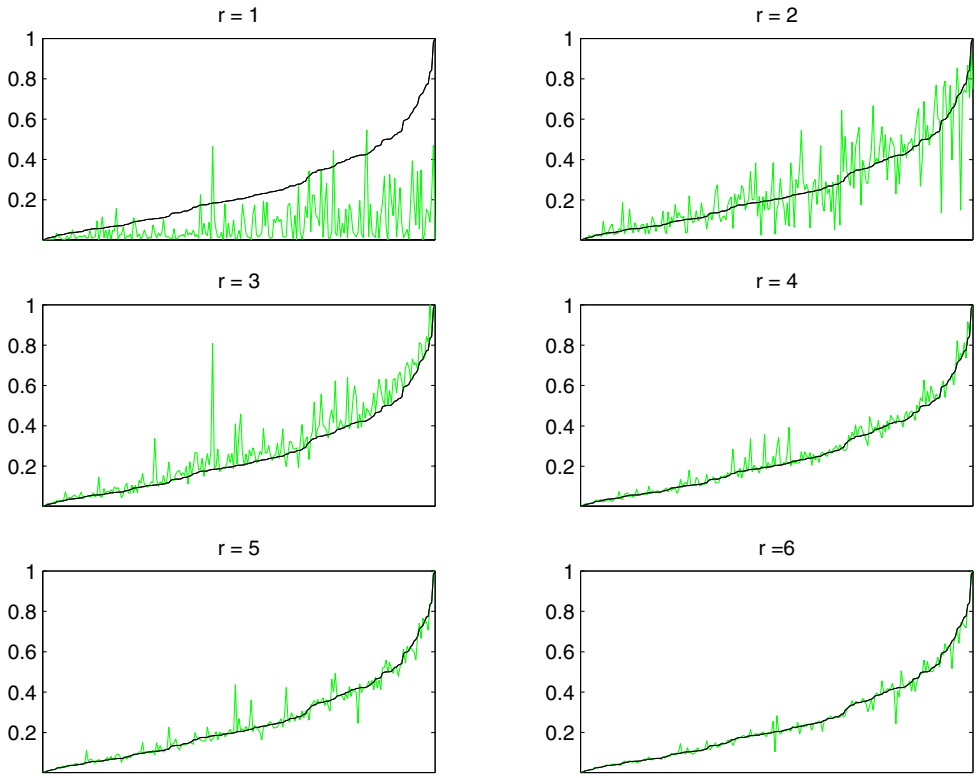


FIG. 3.1. *Average precision (apr) for all* 225 *queries using the Cranfield set for* $r = 1, 2, \ldots, 6$ *in the* BIDIAG *algorithm. The dark lines are the vector model, and the dotted lines are the Krylov subspace model. Queries are sorted after increasing vector model apr.*

The *normal equation residual* $A^T d^{(k)} = A^T(q - \hat{q}^{(k)})$ to the problem (3.11) will tend to zero as k grows. If the normal equation residual converges monotonously to zero,[2] then it is not surprising that the average precision for the Krylov method, using the query expansion scoring $\mathrm{Cqe}_j^{(k)}$ of (3.10), tends to the scoring of the vector model. This is precisely what we see in Figure 3.1. Note that, even if the convergence of $A^T d^{(k)}$ is monotonous, the convergence for the average precision does not have to be monotonous. Looking closely at Figure 3.1, a few such examples are visible.

Finally $d^{(k)}$, the distance between the query and its projection and the normal equation residual $A^T d^{(k)}$, can easily be computed for each step k in the bidiagonalization procedure.

In step k the distance between the query q and the projected query $\hat{q}^{(k)}$ is

$$\begin{aligned} d^{(k)} &= q - \hat{q}^{(k)} \\ &= Q_{k+1}e_1 - Q_{k+1}H_{k+1,k}H_{k+1,k}^T e_1 \\ &= Q_{k+1}(I - H_{k+1,k}H_{k+1,k}^T)e_1 \\ &= Q_{k+1}h_{k+1}^{(k)}h_{k+1}^{(k)T}e_1 \\ &= Q_{k+1}h_{k+1}^{(k)}h_{1,k+1}^{(k)}, \end{aligned} \tag{3.12}$$

[2]The convergence of the normal equation residual is not in general monotonous. For all tests that we made, however, the convergence was monotonous for at least the first 10 iterations.

and its norm is just

$$\|d^{(k)}\| = |h^{(k)}_{1,k+1}|. \tag{3.13}$$

The normal equation residual is

$$\begin{aligned} A^T d^{(k)} &= A^T Q_{k+1} h^{(k)}_{k+1} h^{(k)}_{1,k+1} \\ &= P_{k+1} B^T_{k+1,k+1} h^{(k)}_{k+1} h^{(k)}_{1,k+1} \\ &= P_{k+1} \begin{pmatrix} B^T_{k+1,k} \\ 0 \quad \alpha_{k+1} \end{pmatrix} h^{(k)}_{k+1} h^{(k)}_{1,k+1} \\ &= P_{k+1} \begin{pmatrix} 0 \\ \alpha_{k+1} h^{(k)}_{k+1,k+1} \end{pmatrix} h^{(k)}_{1,k+1}. \end{aligned} \tag{3.14}$$

Its norm is

$$\|A^T d^{(k)}\| = |\alpha_{k+1} h^{(k)}_{k+1,k+1} h^{(k)}_{1,k+1}|. \tag{3.15}$$

3.3. Complexity of the algorithm. In the BIDIAG algorithm, the matrix vector multiplications are performed between a sparse matrix and a dense vector. The number of operations needed is proportional to the number of nonzero elements in A. The rest of the algorithm consists of subtracting and normalizing vectors of length m. In exact arithmetic we will have $Q^T_{r+1} Q_{r+1} = I$ and $P^T_r P_r = I$ (3.1). In standard floating point arithmetic, fully accurate orthogonality of these vectors is observed only at the beginning of the process. In order to recover the orthogonality some type of reorthogonalization would be necessary. This would of course add operations to the complexity of the algorithm. Since we keep the number of iterations r very small, we believe that no reorthogonalization is needed. The main computational work for the document scoring (3.9), (3.10) again is in the size of multiplying a sparse matrix by a dense vector.

4. Numerical experiments.

Data sets. Each one of the test collections we have used consists of a document data base and a set of queries for which relevance judgments are available.

For illustration and comparison purposes, we have used the small and widely circulated data sets Medline, Cranfield, ADI, and CICI.

We have also used larger test collections received from a recent Text Retrieval Conference (TREC) [11]. The TREC 4 disc contains three data collections, the *Financial Times* 1991–1994 (FT), the *Federal Register* 1994 (FR94), and the *Congressional Record* 1993 (CR). The FT collection, FR94 collection, and CR collection consist of 210,158, 55,630, and 27,922 documents, respectively.

Tests on data from the Cranfield collection and from the FT collection will be reported here. Similar tests have been made for the Medline, ADI, CICI, and CR collections. See reports in [4, 5]!

Parsing the data sets. For both collections, any nonzero length string of characters, delimited by white space or a return, was regarded as a term. All terms that occurred in more than 10% of the documents were removed; they were considered to be common words of no interest for the retrieval. Each element $a_{i,j}$ in the term-document matrix was set to the number of occurrences of term number i in document j.

The size of the Cranfield matrix is $7,776$ terms $\times\, 1,400$ documents. Before starting the bidiagonalization process, first the rows and then the columns of the term-document matrix were normalized. This tends to deemphasize common terms and long documents.

The FT term-document matrix is of size $m = 343{,}578$ terms $\times\ n = 210{,}158$ documents with 26,790,949 nonzero elements. The columns were normalized before the bidiagonalization algorithm BIDIAG was started.

Results for the Cranfield collection. There are 225 queries supplied with the test matrix, together with indices j of relevant documents for each query. This gives between 2 and 40 relevant documents for each query; 476 documents were not relevant to any of the queries, 417 documents were relevant to just one, while the remaining 507 documents were relevant to more than one and at most 8 of the 225 queries. We compare our results to these correct answers.

We first summarize the performance in an averaged precision-recall graph. In Figure 4.1 the vector model is compared to LSI and our algorithm, as described in section 3, run for $r = 3$ steps. For the LSI method the optimal rank $s = 296$ in the low rank approximation of A (1.2) was obtained by computing the sum of the average precisions for each query and simply picking the s with the largest sum. It is clear that our Krylov algorithm gives the best averaged precision at all recall levels for these Cranfield data.

Let us look into the details and follow the Golub–Kahan algorithm on one query. Take query 1: it has 29 relevant documents, which is rather many for a Cranfield query. Our algorithm scores this query reasonably well. In Figure 4.2 we follow the progress in linear algebra terms as we execute the algorithm for steps $k = 1, \ldots, 12$. Circles are the residual norms $\|r^{(k)}\|$ (see (3.13)); they decrease unnoticeably slowly from 1 to 0.879. This means that the query q is at a rather large angle to the reached subspace (3.4); it has a projection of length 0.477. We plot the normal equation residuals $\|A^T r^{(k)}\|$ (see (3.15)), as pluses, and note that they decrease reasonably fast at a linear rate. After 12 steps we have found the projection of the query into the document space spanned by A to nearly 3 decimals.

We were curious to see how the singular values converged, and so we plotted estimates of their accuracies as points. Note that the leading singular value converged very quickly; after 12 steps its vector is accurate to 9 decimals, and the singular value to full machine precision. It is well known that the basis vectors Q_k remain orthogonal until one of the singular values converges. We plotted the orthogonality of each basis vector q_k to its predecessors Q_{k-1} as crosses and, true to theory, the crosses and points intersect at half the machine accuracy level, 10^{-8}, during step 10.

Let us now turn to a view of all the documents, and see how well we find the relevant documents for query 1. We plot them in a two-dimensional coordinate system in Figure 4.3. The x-axis is along the projected query $\hat{q}$ (3.7). The y-axis is used to plot the component of each a_j in the reached subspace (3.8) orthogonal to $\hat{q}$. This makes up two of the three components of each a_j vector. We can infer the length of the third component, which is orthogonal to the reached subspace, by remembering that all vectors a_j were normalized to unit length, so the distances of the points plotted to the origin indicate how close the vectors are to the reached subspace. Those shown close to the origin are far from the reached subspace. If we continue the bidiagonalization to full length $r = n$, most of the vectors will get unit length, because then the reached subspace is the whole span of A, except in the rare case when the query is totally unrelated to a part of the document collection.

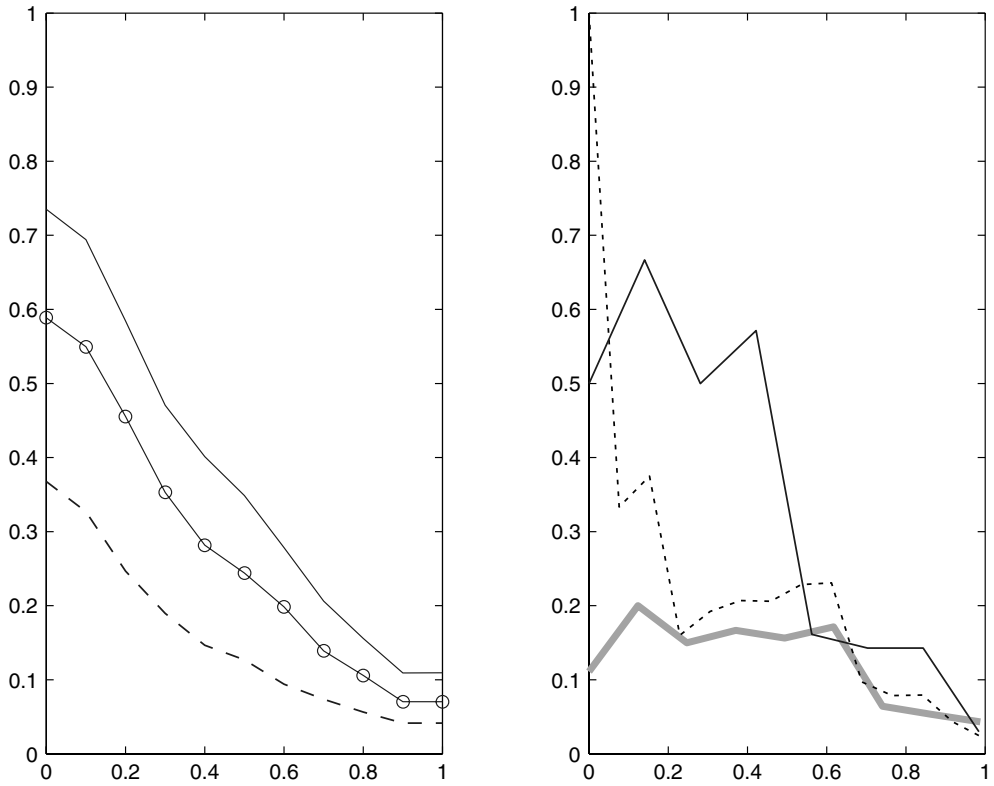


FIG. 4.1. *Precision as a function of recall for the Cranfield collection. Left: Interpolated and averaged over all queries (recall level precision average). Dashed (- -) is the vector model, line with circles (-o) is LSI for rank* $s = 296$*, and plain line (-) is our Krylov algorithm for* $r = 3$ *steps. Right: Our Krylov algorithm to* $r = 3$ *for three different queries, precision at actual recall levels.*

If we use our standard query expansion-based scoring method (3.10), taking angles between the original documents and the projected query, we would choose documents from right to left as plotted in Figure 4.3, and we can check how well we find the relevant documents. We show this by giving the ranking beside each of the ten highest scored relevant documents. Look at the lower part of Figure 4.3, which shows the situation after $r = 2$ steps. First come documents 1, 2, and 3; they are all relevant. Then the next relevant document is retrieved as number 6; we see two nonrelevant documents as points above and close below the circle with number 6. Then the next relevant document is retrieved as number 9. Now our algorithm has given us 10 suggestions, of which we find that 5 are relevant. We say that DLA(10), the document level average precision after 10 documents, is 0.5. The average precision over all relevant documents [11] is lower, 0.297, since the last relevant documents are found much later; we see that the 10th relevant document scores as number 30, while the 29th and last one does not appear until 1029.

Look at the upper half of Figure 4.3, the final one after $r = 12$ steps. There are many points along the y-axis; they denote documents that are orthogonal to the projected query, and will be the last ones scored. Actually 933 of the 1400 documents are orthogonal to the original query.

When scoring documents by angles in the reached plane (3.9), these can be seen as

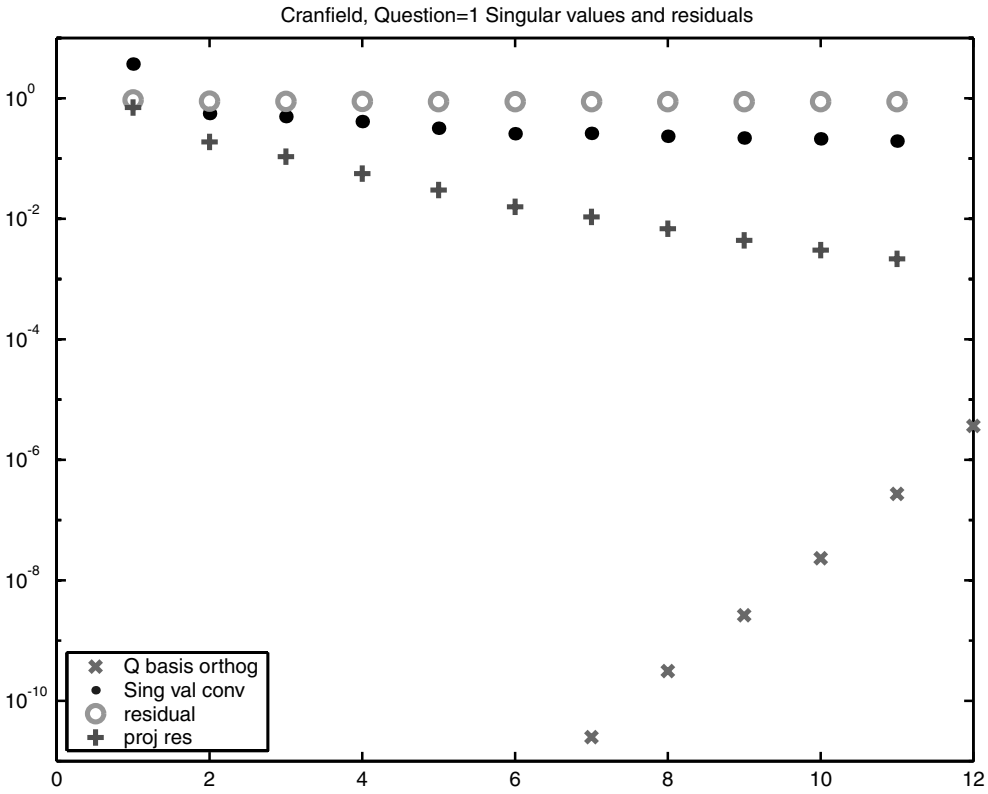


FIG. 4.2. *Convergence of the bidiagonalization procedure starting at query q_1 for the Cranfield matrix.*

angles to the x-axis in Figure 4.3. This scoring did not differ much from the standard query expansion scoring (3.10); for some queries it was better, and for others it was worse. For Query 1, it gave about the same average precision at 0.296 and retrieved relevant documents ranked as 1,2,3,4,5,9, giving a DLA(10) = 0.6. The third scoring choice (angles to Krylov subspace) amounts to choosing those documents plotted far from the origin in Figure 4.3, and gives about the same choices but with lower average precision, 0.180, and DLA(10) = 0.4.

Results for the FT collection. There are several queries provided with the TREC collection. We have used queries 251 to 350. Nine of the queries do not have any relevant answers among the FT documents, and for the rest of the queries there are between 1 and 280 relevant documents. Altogether 3,044 of the 210,158 documents are relevant to some query, 116 documents are relevant to two queries, and 7 documents are relevant to three queries.

In Figure 4.4 the vector model is compared to our algorithm run to $r = 3$. The experiments were made in the same way as for Figure 4.1, but we did not have results for LSI for this large matrix. Documents were scored using the standard query expansion scores (3.10). We did choose $r = 3$ as dimension of the Krylov subspace; here the results were better for larger subspaces for some of the queries.

We choose such a query, number 344, to report in Figure 4.5. As for Figure 4.3, the x-axis is along the projected query $\hat{q}$ (3.7), and the y-axis is used to plot the

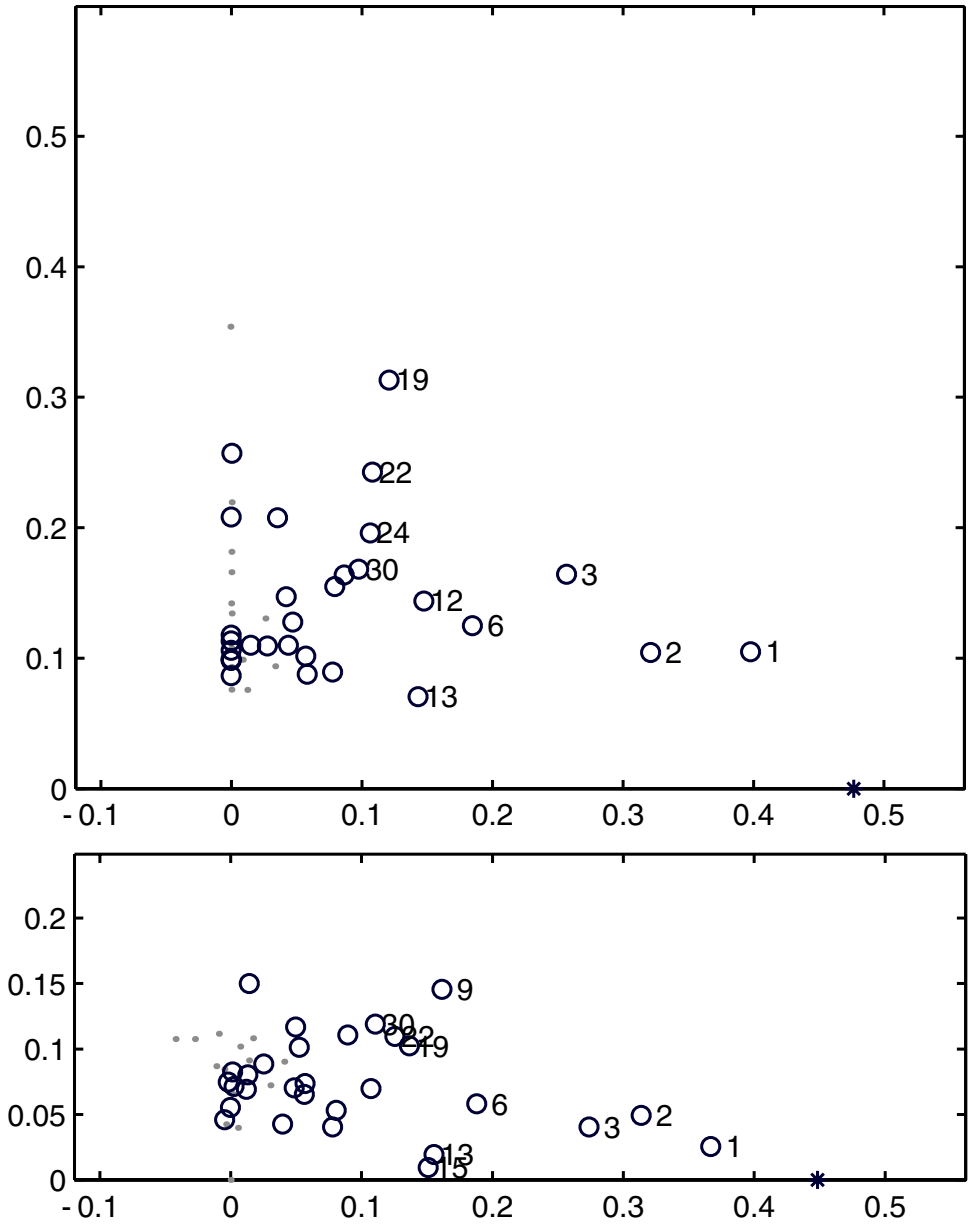


FIG. 4.3. *Results for the Cranfield matrix for Query* 1 *at steps* $r = 12$ *(upper panel) or* $r = 2$ *(lower panel). Numbers are rankings given by the algorithm to relevant documents. Circles mark relevant documents, while points mark those not relevant. An asterisk marks the projected query. Horizontal (x-axis): component in direction of the projected query* $\hat{q}$*; vertical (y-axis): component orthogonal to* $\hat{q}$ *in reached subspace.*

component of each document vector in the reached subspace. The labels show the ranking of the relevant documents; there are only 3 relevant documents among all the 210,158, quite like seeking a needle in a haystack. Note that the relevant documents

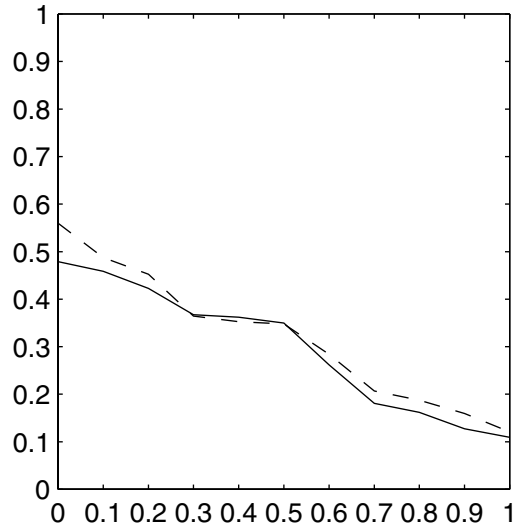


FIG. 4.4. *Interpolated precisions for recall levels* $0, 0.1, \ldots, 1$ *for the FT collection from the TREC data base. The vector model (–) is compared to our algorithm for* $r = 3$ *(- -). The average of the* 25 *documents that are best ranked by the vector space method is included.*

get better ranking for the larger subspace $r = 6$ than for $r = 3$. This question is not one of the 25 best questions included in Figure 4.4.

Discussion. The experiments have shown good performance for the small data set (Cranield) but not very good performance for the larger FT set. Although we cannot notice any major differences in the structure of the term-document matrices or the distribution of singular values, there are differences between the two sets. The FT set consists of news telegrams and Cranfield of scientific papers. For the Cranfield collection, most users will probably agree on the relevance judgments given for this set, while for the FT documents more subjectivity is involved in the relevance judgments. We believe the larger sets do reflect a more realistic case.

The construction of the FT matrix also plays a role in the performance of our algorithm. Perhaps more care has to be taken when deciding what terms to use for the matrix. It might not be enough to remove all terms occurring in more than 10% of the documents; maybe that figure should be 5% or something else.

Some type of row and column normalization is useful. In our Cranfield experiments, we first normalized the row vectors, and then the column vectors. Even if the normalization of the column vectors destroys the row normalization, a smoothing effect remains. This had some effect on the performance for the Cranfield matrix. For the FT matrix only the columns were normalized.

The starting vector (the query) in our algorithm plays an important role, and it might also benefit our algorithm to pay more attention to how to construct the query vector. We have only tried our algorithm for at most $r = 12$ steps, since generating a larger subspace is too time-consuming to be interesting in a realistic case. Moreover, the starting vector loses its importance the longer we iterate. For our future work we will concentrate on improving the starting vector, and we will investigate how to add relevance feedback to the algorithm.

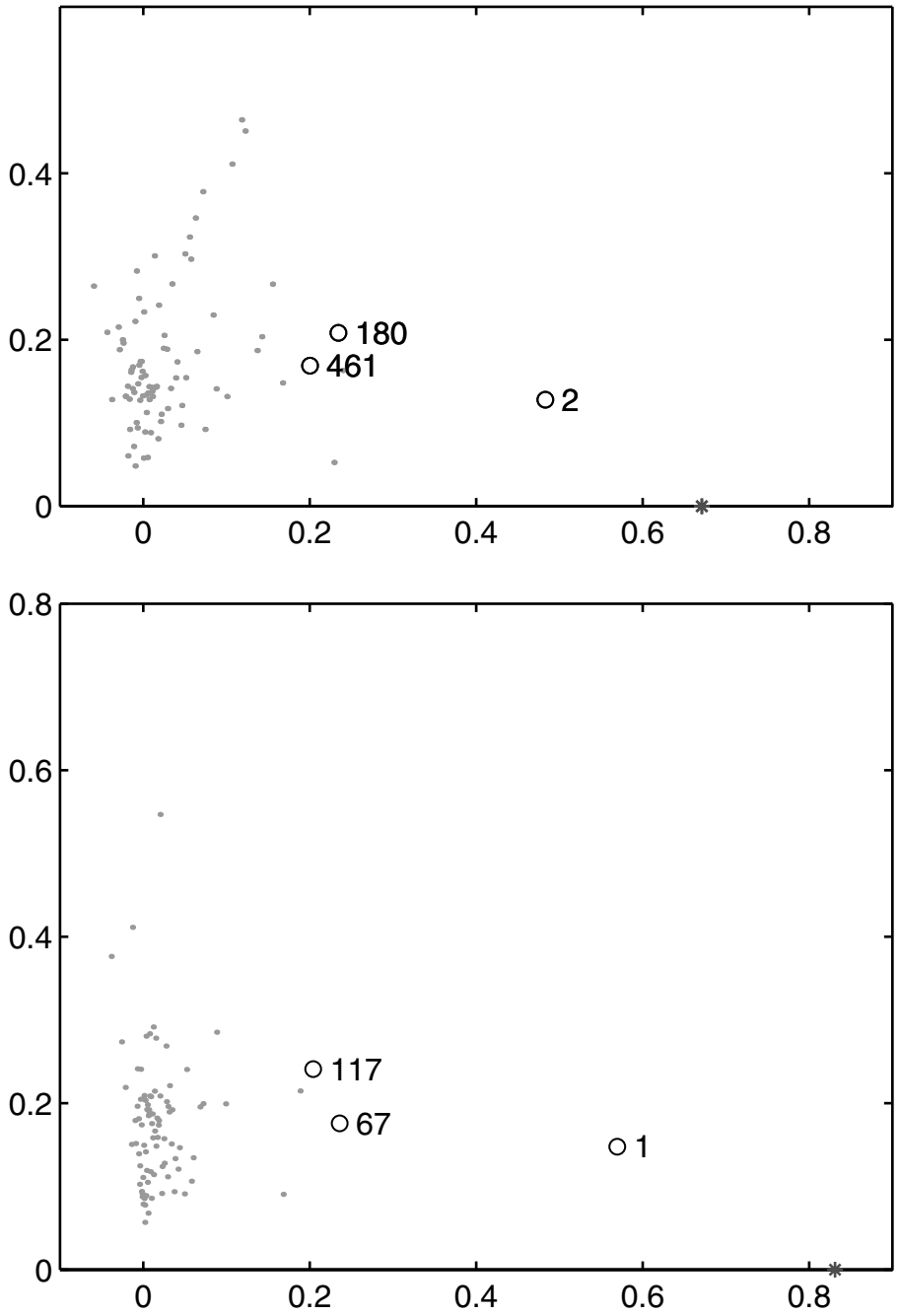


FIG. 4.5. *Results for the TREC FT matrix for Query* 344*, at steps* $r = 3$ *(upper panel) and* $r = 6$ *(lower panel). Numbers indicate rankings of relevant documents. For the upper panel,* 97% *of the documents are in the interval* < 0.1*, and for the lower panel,* 99% *of the documents are in that interval; only a sample of those are shown. An asterisk marks the projected query.*

REFERENCES

[1] 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.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, *Latent Dirichlet allocation*, J. Mach. Learn. Res., 3 (2003), pp. 993–1022.

[3] K. Blom, *Information Retrieval Using the Singular Value Decomposition and Krylov Subspaces*, Technical report 1999-5, Department of Mathematics, Chalmers University of Technology, Göteborg, Sweden, 1999.

[4] K. Blom, *Information Retrieval Using Krylov Subspace Methods*, Ph.D. thesis, Department of Mathematics, Chalmers University of Technology, Göteborg, Sweden, 2004.

[5] K. Blom and A. Ruhe, *Information retrieval using very short Krylov sequences*, in Computational Information Retrieval, M. W. Berry, ed., Proc. Appl. Math. 106, SIAM, Philadelphia, 2001, pp. 39–52.

[6] I. S. Dhillon and D. S. Modha, *Concept decompositions for large sparse text data using clustering*, Machine Learning, 42 (2001), pp. 143–175.

[7] S. T. Dumais, *Latent semantic indexing (LSI): TREC-3 report*, in Proceedings of the Third Text REtrieval Conference (TREC-3), NIST Special Publication 500-225, D. K. Harman, ed., NIST, Gaithersburg, MD, 1995, pp. 219–230.

[8] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman, *Indexing by latent semantic analysis*, J. Amer. Soc. Inform. Sci., 41 (1990), pp. 391–407.

[9] G. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM J. Numer. Anal., 2 (1965), pp. 205–224.

[10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[11] D. Harman, *The Eighth Text REtrieval Conference (TREC-8)*, NIST Special Publication 500-246, NIST, Gaithersburg, MD, 2000; also available online at http://trec.nist.gov/pubs/trec8.

[12] T. Hofmann, *Probabilistic latent semantic indexing*, in Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, University of California, Berkeley, ACM Press, New York, 1999, pp. 50–57.

[13] H. Park, M. Jeon, and J. B. Rosen, *Lower dimensional representation of text data in vector space based information retrieval*, in Computational Information Retrieval, M. W. Berry, ed., Proc. Appl. Math. 106, SIAM, Philadelphia, 2001, pp. 3–23.

[14] K. Sparck Jones, *Automatic Keyword Classification for Information Retrieval*, Butterworth, London, 1971.

[15] J. Xu and W. B. Croft, *Query expansion using local and global document analysis*, in Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland, ACM Press, New York, 1996, pp. 4–11.

[16] H. Zha, O. Marques, and H. D. Simon, *Large-scale SVD and subspace-based methods for information retrieval*, in Solving Irregularly Structured Problems in Parallel, A. Ferreira, J. Rolim, H. Simon, and S. Teng, eds., Lecture Notes in Comput. Sci. 1457, Springer, New York, 1998, pp. 29–42.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 2, pp. 583–598

AN EFFICIENT APPROACH TO THE LINEAR LEAST SQUARES PROBLEM*

K. TUNYAN†, K. EGIAZARIAN†, A. TUNIEV‡, AND J. ASTOLA†

Abstract. In this paper we present a partially orthogonal decomposition for a matrix A. Using this decomposition the linear least squares problem is reduced to solving two linear systems. The matrix of the first system is symmetric and positive definite, and the matrix of the second system is nonsingular upper triangular. We show that this approach can provide computational savings.

Key words. orthogonalization process, QR decomposition, pseudoinverses

AMS subject classifications. 65F25, 15A23, 15A09, 65F20, 65F05

DOI. 10.1137/S0895479801386596

1. Introduction.

One of the key elements in solving practical problems is construction of computationally efficient and stable algorithms. Different matrix factorizations are widely used for this purpose. There are many known matrix decompositions, and the list is growing. A powerful tool allowing one to obtain various factorizations of a matrix was described in [4], where different matrix factorizations have been unified.

For solving linear least squares problems the QR decomposition is often used and there are different methods for computing the QR factorization [1, 2, 4, 5, 7, 8, 9, 11, 14, 15].

In this paper, we suggest such an approach to solving the linear least squares problem that can provide computational savings in comparison with the algorithms using modified Gram–Schmidt orthogonalization and Householder transformations. The key idea here is to create a special decomposition of the matrix by utilizing a partial orthogonalization process proposed in 1980 [18], which is indeed a special case of the generalized process described in 1995 [4].

In [18, 19] it was proposed to choose k elements simultaneously—the pivot vector of length k—instead of the pivot element in the Gauss transformation of the elimination method, where $k \in \{1, \dots, n\}$, n is the number of columns of a matrix A. This idea leads to a parametric linear transformation depending on k, which in geometrical terms is a realization of the idea of partial orthogonalization and in algebraic terms it is a "convex" combination of the Gauss transformation of the elimination method ($k = 1$) and the Gram–Schmidt transformation of the orthogonalization process ($k = n$). Using this transformation the parametric partial orthogonalization process was obtained (see, for example, [19]).

This paper is organized as follows. Section 2 describes the parametric linear transformation, the partial orthogonalization process, and its modified version. The stability analysis of the modified partial orthogonalization process is also given. It is shown that the modified version provides computational stability if the growth

*Received by the editors May 3, 2001; accepted for publication (in revised form) by M. Chu January 13, 2003; published electronically January 12, 2005.
http://www.siam.org/journals/simax/26-2/38659.html
†Institute of Signal Processing, Tampere University of Technology, P.O. Box 553, FIN-33101, Tampere, Finland (tunyan@cs.tut.fi, karen@cs.tut.fi, jta@cs.tut.fi).
‡Institute for Informatics and Automation Problems, National Academy of Sciences of Armenia and Yerevan State University, P. Sevak 1, 375014, Yerevan, Armenia.

factor is not large. Section 3 presents the generalization of QR decomposition by performing the modified partial orthogonalization process and section 4 uses this development to calculate the normal pseudosolution by the obtained decomposition. In section 5 the block approach to this problem is described and section 6 illustrates how to calculate normal pseudosolution of the linear least squares problem in this case. The concepts presented in sections 2–6 are illustrated by some numerical examples. Section 7 gives the number of required arithmetical operations for solving the least squares problem in two cases (general and block approaches). It is shown that in the block case the proposed method can be more effective at the certain values of parameters than the algorithms using modified Gram–Schmidt orthogonalization and Householder transformations.

2. Partial orthogonalization process. Let $N = \{1, \dots, n\}$, $M = \{1, \dots, m\}$. By $x[K]$ we denote the corresponding K-piece of the vector $x \in \mathbb{R}^n$, where $K \subset N$. We do not distinguish between row and column vectors.

2.1. Parametric linear transformation. Let a system of vectors $a_1, \dots, a_m \in \mathbb{R}^n$ be given. Let $K \subset N$ and the Euclidean norm of $a_1[K]$ be $\|a_1[K]\| \neq 0$. Set

$$\left.\begin{aligned} b_1 &= \frac{a_1}{\|a_1[K]\|}, \\ b_i &= a_i + \alpha_i b_1 \quad \text{for all } i > 1 \end{aligned}\right\},$$

where $\alpha_i = \alpha_i(K) = -a_i[K]b_1[K]$, $i > 1$.

Let $K = \{1, \dots, k\}$, where $k \leq n$. This transformation is a generalization of both the Gauss transformation (for $k > 1$) and the Gram–Schmidt transformation (for $k < n$). Note that for all $i > 1$

$$b_i[K]b_1[K] = (a_i[K] + \alpha_i b_1[K])b_1[K] = -\alpha_i + \alpha_i = 0;$$

that is, the subvectors $b_i[K]$ are orthogonal to $b_1[K]$.

The vectors a and $b \in \mathbb{R}^n$ are called *partially orthogonal* if $a[K]b[K] = 0$, $K \subset N$, and are called *partially orthonormal* if the norms of these subvectors are equal to 1.

2.2. Partial orthogonalization process and its modification. Consider a system of vectors $a_1, \dots, a_m \in \mathbb{R}^n$, where $m \leq n$. Let $K \subset N$ and the number of elements in the set K be equal to k, $k \geq m$. Without loss of generality, we assume that $K = \{1, \dots, k\}$, and the subvectors $a_1[K], \dots, a_m[K] \in \mathbb{R}^k$ are linearly independent. Then the partial orthogonalization process permits one to obtain from the system of vectors $a_1, \dots, a_m$ another system of linearly independent vectors $b_1, \dots, b_m \in \mathbb{R}^n$, where the subvectors $b_1[K], \dots, b_m[K] \in \mathbb{R}^k$ are orthonormal.

The system of vectors $b_1, \dots, b_m$ is obtained in the following way. Successively for $s = 1, \dots, m$

$$\left.\begin{aligned} b_s &= \frac{\overline{b}_s}{\|\overline{b}_s[K]\|} \qquad (\overline{b}_1 = a_1), \\ \overline{b}_s &= a_s + \alpha_1 b_1 + \cdots + \alpha_{s-1} b_{s-1}, \end{aligned}\right. \tag{2.1}$$

where

$$\alpha_i = \alpha_i(K) = -a_s[K]b_i[K], \qquad i = 1, \dots, s-1. \tag{2.2}$$

If $K = N$ in (2.1) and (2.2), then we obtain the Gram–Schmidt orthogonalization; that is, the system of vectors $b_1, \dots, b_m$ is orthogonal (orthonormal).

If $K \subset N$, then the process (2.1)–(2.2) is called a *partial orthogonalization process* and the system of vectors $b_1, \dots, b_m$ is called *partially orthogonal*.

The number of arithmetical operations (1 operation = 1 addition + 1 multiplication) for the partial orthogonalization process is $(k+n)m^2/2$, where $k \in \{m, m+1, \dots, n\}$. If $k = n$, then we obtain the number of operations for the orthogonalization process, that is, nm^2.

Further we will use the modified partial orthogonalization process, where at each sth step we transform not only one row, but all the rows which are below the sth row. At the end of this process we obtain the system of partially orthogonal vectors. To obtain a partially orthonormal system we partially orthonormalize this system.

Let us consider the modified partial orthogonalization process. Denote

$$a_i^1 = a_i, \qquad i = 1, \dots, m,$$

and further successively for $s = 1, \dots, m-1$ set

$$a_i^{s+1} = a_i^s + \alpha_i a_s^s, \qquad i = s+1, \dots, m,$$

where

$$\alpha_i = \alpha_i^s(K) = -\frac{a_i^s[K]a_s^s[K]}{a_s^s[K]a_s^s[K]}, \qquad i = s+1, \dots, m.$$

After m steps we obtain the system of partially orthogonal vectors $a_1^1, \dots, a_m^m$. Partially normalizing these vectors, that is, by setting

$$q_s = \frac{a_s^s}{\|a_s^s[K]\|}, \qquad s = 1, \dots, m,$$

we obtain the system of partially orthonormal vectors $q_1, \dots, q_m$.

Note that if we obtain the system $\{b_s\} \in \mathbb{R}^n$ by the partial orthogonalization process, then $b_s = q_s$ for $s = 1, \dots, m$.

Example. Let us consider the modified partial orthogonalization process on the next example. Let $m = 3$, $n = 6$, and the vectors

$$a_1 = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1), \quad a_2 = (1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1), \quad a_3 = (0 \quad 0 \quad 1 \quad 1 \quad -1 \quad 0)$$

be given.[1] Select $K = \{1, 2, 3, 4\}$. Then after each of three steps we obtain Tables 2.1–2.3, respectively.

Denote

$$a_1^1 = (1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1), \; a_2^2 = \left(\frac{3}{4} \;\; -\frac{1}{4} \;\; -\frac{1}{4} \;\; -\frac{1}{4} \;\; -\frac{1}{4} \;\; \frac{3}{4}\right),$$
$$a_3^3 = \left(0 \;\; -\frac{2}{3} \;\; \frac{1}{3} \;\; \frac{1}{3} \;\; -\frac{5}{3} \;\; 0\right).$$

Since the norms $\|a_1^1[K]\| = 2$, $\|a_2^2[K]\| = \frac{\sqrt{12}}{4}$, $\|a_3^3[K]\| = \frac{\sqrt{6}}{3}$, the vectors

$$q_1 = \left(\frac{1}{2} \;\; \frac{1}{2} \;\; \frac{1}{2} \;\; \frac{1}{2} \;\; \frac{1}{2} \;\; \frac{1}{2}\right), \; q_2 = \left(\frac{3}{\sqrt{12}} \;\; -\frac{1}{\sqrt{12}} \;\; -\frac{1}{\sqrt{12}} \;\; -\frac{1}{\sqrt{12}} \;\; -\frac{1}{\sqrt{12}} \;\; \frac{3}{\sqrt{12}}\right),$$
$$q_3 = \left(0 \;\; -\frac{2}{\sqrt{6}} \;\; \frac{1}{\sqrt{6}} \;\; \frac{1}{\sqrt{6}} \;\; -\frac{5}{\sqrt{6}} \;\; 0\right)$$

are partially orthonormal.

[1] Note that this example will be used again later on.

TABLE 2.1

K				$N\backslash K$		
1	2	3	4	5	6	α_i
1	1	1	1	1	1	−
1	0	0	0	0	1	$-\frac{1}{4}$
0	0	1	1	−1	0	$-\frac{1}{2}$

TABLE 2.2

K				$N\backslash K$		
1	2	3	4	5	6	α_i
1	1	1	1	1	1	−
$\frac{3}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$	$\frac{3}{4}$	−
$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{3}{2}$	$-\frac{1}{2}$	$\frac{2}{3}$

TABLE 2.3

K				$N\backslash K$	
1	2	3	4	5	6
1	1	1	1	1	1
$\frac{3}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$	$\frac{3}{4}$
0	$-\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$-\frac{5}{3}$	0

$A \in \mathbb{R}^{m\times n}$ is called a *partially orthogonal matrix* if its submatrix $A_1 \in \mathbb{R}^{m\times k}$, $k < n$, is a matrix with orthogonal rows.

2.3. On stability of the modified partial orthogonalization process. Several authors have discussed the stability of both the Gaussian elimination and the Gram–Schmidt orthogonalization (see, for example, [2, 3, 5, 6, 7, 8, 9, 11, 13, 14, 15, 20, 21]). Both processes are unstable. Nevertheless, it has been shown that if the procedures are organized in a proper way, then more accurate results are possible. In particular, the Gaussian elimination with partial pivoting provides stable computation if the growth factor is not large (see, for example, [5, 7, 8, 14, 17, 20, 21]).

On the other hand, the modified Gram–Schmidt orthogonalization has significantly different numerical properties than the classical one [5, 7, 8, 11, 15, 20, 21], but both of them require reorthogonalization for stable computations. It was also demonstrated that, for solving the linear least squares problem by the modified version, reorthogonalization is not needed [3]. Besides, the stability of the modified version is independent of the pivoting strategies, even though they can readily be adopted to this version [3, 13].

We propose to use the modified partial orthogonalization process with the following pivoting rule. At each sth step ($s = 1, \dots, m$) as a pivot row we choose the ith row, where $\|a_i^s[K]\|$, $s \le i \le m$, is maximized. Now we can guarantee that the absolute values of all multipliers α_i^s are bounded by 1. Therefore, each element of the current matrix may at most double. Indeed,

$$|a_{ij}^s| = |a_{ij}^{s-1} + \alpha_i^s a_{sj}^{s-1}| \le |a_{ij}^{s-1}| + |a_{sj}^{s-1}| \le 2 \max_{i,j} |a_{ij}^{s-1}|, \qquad i, j > s.$$

A growth factor for this method can be calculated similarly to the one for the

Gaussian elimination, that is,

$$\rho = \frac{\max\limits_{i,j,s} |a_{ij}^s|}{\max\limits_{i,j} |a_{ij}|} = 2^{\tau-1},$$

which in the extreme case $\tau = n$ coincides with the estimation obtained by Wilkinson [21]. Since the growth factor for the proposed method with suggested pivoting rule is bounded, we can conclude that the method is stable for a given pivoting strategy if the growth factor is small, which is usually the case in the practical applications.

For the pivoting strategy it is required to perform additionally mk operations and m comparisons.

To ensure stability it is convenient to assume hereafter that the "modified partial orthogonalization process" means the "modified partial orthogonalization process with the proposed pivoting rule."

3. Development of QR decomposition. Let $A \in \mathbb{R}^{m\times n}$, $m \le n$, be a full rank matrix. It is well known that in this case using the orthogonalization process the matrix A can be represented in the form $A = RQ$, where $R \in \mathbb{R}^{m\times m}$ is a nonsingular lower triangular matrix and $Q \in \mathbb{R}^{m\times n}$ is a matrix with orthogonal (orthonormal) rows.

Using the modified partial orthogonalization process we now generalize the QR decomposition.

THEOREM 3.1. *Let the rank of the submatrix $A_1 \in \mathbb{R}^{m\times k}$, $k \ge m$, of the matrix $A \in \mathbb{R}^{m\times n}$ be equal to m. Then*

$$A = RQ = R(Q_1, Q_2), \tag{3.1}$$

where $R \in \mathbb{R}^{m\times m}$ is a nonsingular lower triangular matrix and $Q = (Q_1, Q_2) \in \mathbb{R}^{m\times n}$ is a partially orthogonal matrix; that is, $Q_1 \in \mathbb{R}^{m\times k}$ is a matrix with orthogonal rows.

Proof. Denote $A^1 = A$. Construct the matrix

$$M^1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \alpha_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m1} & 0 & \cdots & 1 \end{bmatrix},$$

where $\alpha_{i1} = \alpha_i^1(K) = -\frac{a_i^1[K]a_1^1[K]}{a_1^1[K]a_1^1[K]}$, $i = 2, 3, \ldots, m$.

Since we assume $A_1 \in \mathbb{R}^{m\times k}$ to be a full rank matrix, $a_1^1[K]$ will be a nonzero vector. Define the matrix

$$A^2 = M^1 A^1. \tag{3.2}$$

Note that the matrix A^2 is obtained from the matrix A^1 by the modified partial orthogonalization (first step, $s = 1$).

Similarly, successively for $s = 2, 3, \ldots, m-1$ we construct the matrices

$$M^s = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & \alpha_{s+1,s} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \alpha_{ms} & \cdots & 1 \end{bmatrix}, \tag{3.3}$$

where $\alpha_{is} = \alpha_i^s(K) = -\frac{a_i^s[K]a_s^s[K]}{a_s^s[K]a_s^s[K]}$, $i = s+1, \ldots, m$, and find

$$A^{s+1} = M^s A^s. \tag{3.4}$$

For $s = m-1$

$$A^m = M^{m-1}A^{m-1},$$

where the matrix $A^m \in \mathbb{R}^{m\times n}$ is partially orthogonal, since its submatrix $A_1^m \in \mathbb{R}^{m\times k}$ is a matrix with orthogonal rows. From (3.2) and (3.4) we can see that

$$A^m = MA^1 = MA, \tag{3.5}$$

where $M = M^{m-1}M^{m-2}\ldots M^1$. Since the matrices M^s have the form (3.3) for $s = 1, \ldots, m-1$, then we obtain the inverses of these matrices by reversing the signs of their underdiagonal elements. Therefore, the inverse of M is

$$\widetilde{M} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -\alpha_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -\alpha_{m1} & -\alpha_{m2} & \ldots & 1 \end{bmatrix}.$$

From (3.5) we obtain the decomposition

$$A = \widetilde{M}A^m, \tag{3.6}$$

where $\widetilde{M} \in \mathbb{R}^{m\times m}$ is a nonsingular lower triangular matrix and $A^m \in \mathbb{R}^{m\times n}$ is a partially orthogonal matrix. To partially orthonormalize the rows of A^m we set $q_s = \frac{a_s^m}{\|a_s^m[K]\|}$ and $R = (r_1, \ldots, r_s, \ldots, r_m)$, where $R_s = \widetilde{\alpha}_s\|a_s^m[K]\|$, $s = 1, \ldots, m$ ($\widetilde{\alpha}_s$ are the columns of $\widetilde{M}$). Thus, from (3.6) we obtain the decomposition (3.1), that is, $A = RQ = R(Q_1, Q_2)$. □

If in the matrix $A \in \mathbb{R}^{m\times n}$ the number of rows is greater than the number of columns ($m > n$), then the decomposition of A has the form

$$A = QR = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R, \tag{3.7}$$

where $Q_1 \in \mathbb{R}^{k\times n}$ is a matrix with orthonormal columns and $R \in \mathbb{R}^{n\times n}$ is a nonsingular upper triangular matrix.

If $k = n$ in the decomposition (3.1), then we obtain the QR decomposition of A. If $k < n$, then the decomposition (3.1) is called a *partially orthogonal decomposition of* A, or a $Q_K R_K$ *decomposition*.

Example. Consider the example given for the modified partial orthogonalization process. In this case, according to Theorem 3.1, we obtain

$$A^3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & \frac{3}{4} \\ 0 & -\frac{2}{3} & \frac{1}{3} & \frac{1}{3} & -\frac{5}{3} & 0 \end{bmatrix}, \qquad \widetilde{M} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & -\frac{2}{3} & 1 \end{bmatrix}.$$

Therefore,

$$Q = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{3}{\sqrt{12}} & -\frac{1}{\sqrt{12}} & -\frac{1}{\sqrt{12}} & -\frac{1}{\sqrt{12}} & -\frac{1}{\sqrt{12}} & \frac{3}{\sqrt{12}} \\ 0 & -\frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{5}{\sqrt{6}} & 0 \end{bmatrix} \text{ is a partially orthonormal matrix,}$$

$$R = \begin{bmatrix} 2 & 0 & 0 \\ \frac{1}{2} & \frac{\sqrt{12}}{4} & 0 \\ 1 & -\frac{\sqrt{12}}{6} & \frac{\sqrt{6}}{3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & -\frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & \frac{\sqrt{12}}{4} & 0 \\ 0 & 0 & \frac{\sqrt{6}}{3} \end{bmatrix}.$$

It is easy to see that $A = RQ$, that is,

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ \frac{1}{2} & \frac{\sqrt{12}}{4} & 0 \\ 1 & -\frac{\sqrt{12}}{6} & \frac{\sqrt{6}}{3} \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{3}{\sqrt{12}} & -\frac{1}{\sqrt{12}} & -\frac{1}{\sqrt{12}} & -\frac{1}{\sqrt{12}} & -\frac{1}{\sqrt{12}} & \frac{3}{\sqrt{12}} \\ 0 & -\frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{5}{\sqrt{6}} & 0 \end{bmatrix}. \tag{3.8}$$

4. Calculation of normal pseudosolution by partially orthogonal decomposition. Let the rank of $A \in \mathbb{R}^{m\times n}$, $m > n$, be equal to n.

Consider the least squares problem

$$Ax \cong b, \tag{4.1}$$

where $x \in \mathbb{R}^n$ is an unknown column. We will show that if the $Q_K R_K$ decomposition (3.7) is known, then the calculation of the normal pseudosolution of the system (4.1) is reduced to the problem of solving the system

$$URx = r,$$

or the following two systems:

$$Uy = r, \tag{4.2}$$

$$Rx = y, \tag{4.3}$$

where $U = Q^TQ = I + Q_2^TQ_2 \in \mathbb{R}^{n\times n}$ is a symmetric and positive definite matrix, $R \in \mathbb{R}^{n\times n}$ is an upper triangular matrix, $r = Q^Tb \in \mathbb{R}^n$, $I \in \mathbb{R}^{n\times n}$ is the identity matrix, and $y \in \mathbb{R}^n$ is an unknown column.

To prove this statement we consider the solvable normal system

$$A^TAx = A^Tb \tag{4.4}$$

instead of (4.1). Let $k < m$ and the $Q_K R_K$ decomposition (3.7) be obtained. Since $Q_1 \in \mathbb{R}^{k\times n}$ is a matrix with orthonormal columns, $Q_1^T Q_1 = I$. Therefore,

$$A^T A = (QR)^T QR = R^T Q^T QR = R^T \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}^T \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R$$

$$= R^T (Q_1^T Q_1 + Q_2^T Q_2) R = R^T (I + D) R = R^T U R.$$

Here $D = Q_2^T Q_2$ is a symmetric matrix, and it is well known that the matrix in the form $U = I + D$ is symmetric and positive definite. Thus

$$A^T A = R^T U R. \tag{4.5}$$

On the other hand, the right side of the normal system (4.4) is

$$A^T b = (QR)^T b = R^T Q^T b = R^T r, \tag{4.6}$$

where $r = Q^T b$.

Taking into account (4.5) and (4.6), we represent the system (4.4) in the form $R^T U R x = R^T r$. By left-multiplying both sides of this system with the inverse of R^T, we obtain $URx = r$.

From here it follows that for the calculation of a normal pseudosolution it is necessary to solve two systems, (4.2) and (4.3).

Remark 4.1. If we select $k = m$, then $U \in \mathbb{R}^{n\times n}$ is an identity matrix and the solution of the system (4.2) is $y = r$. In other words, at $k = m$, the calculation of a normal pseudosolution is reduced to solving only the triangular system (4.3). This result coincides with the result obtained by using the QR decomposition of the orthogonalization process [3, 7].

Example. Let us consider the example of problem (4.1), where A and b are

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & -1 & 0 \end{bmatrix}^T, \qquad b = (\ 1 \quad 1 \quad -1 \quad 1 \quad 1 \quad 1\ )^T.$$

Taking into account (3.7) and the form of decomposition (3.8), we have $A = QR$, where

$$\begin{array}{r} \\ K \\ -- \\ M\backslash K \end{array} \begin{array}{c} A \\ \left[\begin{array}{ccc} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{array}\right] \end{array} = \begin{array}{c} Q \\ \left[\begin{array}{ccc} \frac{1}{2} & \frac{3}{\sqrt{12}} & 0 \\ \frac{1}{2} & -\frac{1}{\sqrt{12}} & -\frac{2}{\sqrt{6}} \\ \frac{1}{2} & -\frac{1}{\sqrt{12}} & \frac{1}{\sqrt{6}} \\ \frac{1}{2} & -\frac{1}{\sqrt{12}} & \frac{1}{\sqrt{6}} \\ \frac{1}{2} & -\frac{1}{\sqrt{12}} & -\frac{5}{\sqrt{6}} \\ \frac{1}{2} & \frac{3}{\sqrt{12}} & 0 \end{array}\right] \end{array} \begin{array}{c} R \\ \left[\begin{array}{ccc} 2 & \frac{1}{2} & 1 \\ 0 & \frac{\sqrt{12}}{4} & -\frac{\sqrt{12}}{6} \\ 0 & 0 & \frac{\sqrt{6}}{3} \end{array}\right], \end{array}$$

or in the block form,

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R.$$

For this example we obtain

$$D = Q_2^T Q_2 = \begin{bmatrix} \frac{1}{2} & \frac{1}{\sqrt{12}} & -\frac{5}{2\sqrt{6}} \\ \frac{1}{\sqrt{12}} & \frac{5}{6} & \frac{5}{6\sqrt{2}} \\ -\frac{5}{2\sqrt{6}} & \frac{5}{6\sqrt{2}} & \frac{25}{6} \end{bmatrix},$$

$$U = I + D = \begin{bmatrix} 1+\frac{1}{2} & \frac{1}{\sqrt{12}} & -\frac{5}{2\sqrt{6}} \\ \frac{1}{\sqrt{12}} & 1+\frac{5}{6} & \frac{5}{6\sqrt{2}} \\ -\frac{5}{2\sqrt{6}} & \frac{5}{6\sqrt{2}} & 1+\frac{25}{6} \end{bmatrix}, \quad r = Q^T b = \begin{bmatrix} 2 \\ \frac{2}{\sqrt{3}} \\ -\frac{7}{\sqrt{6}} \end{bmatrix}.$$

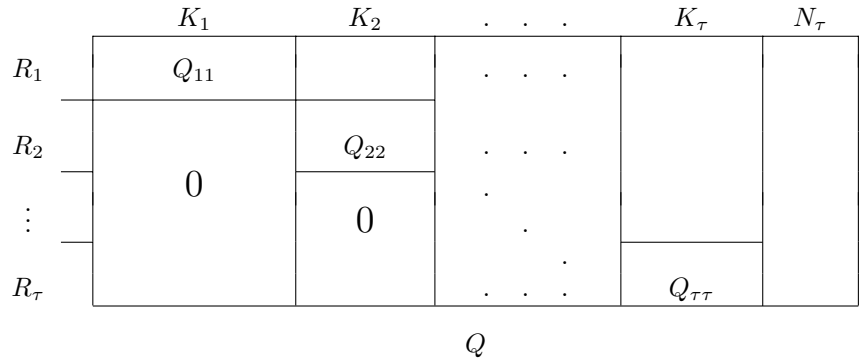
Solving the system (4.2) by the Cholesky method, we obtain $y = (\frac{10}{11} \quad \frac{2\sqrt{12}}{11} \quad -\frac{2\sqrt{6}}{11})^T$. Further, solving the triangular system (4.3), we obtain the normal pseudosolution $x = (\frac{7}{11} \quad \frac{4}{11} \quad -\frac{6}{11})^T$.

5. Further development of QR decomposition (block approach). Consider the matrix $A \in \mathbb{R}^{m\times n}$, $m \le n$. To obtain the $Q_K R_K$ decomposition of the matrix A for the modified partial orthogonalization process we assume that the number of elements of the set K is equal to k, where $k \in \{m, m+1, \dots, n\}$, and the rank of the submatrix $A_1 \in \mathbb{R}^{m\times k}$ is equal to m. Now we remove these restrictions and assume only that the rank of A is equal to m. In this case we consider a new process based on the modified partial orthogonalization process. This process permits one to obtain a decomposition of A in the form

$$A = RQ, \tag{5.1}$$

where $R \in \mathbb{R}^{m\times m}$ is a nonsingular lower triangular matrix, $Q \in \mathbb{R}^{m\times n}$ is a rectangular upper block-diagonal matrix with τ blocks, and each block (not necessarily square) is a matrix with orthogonal rows. In other words, the matrix Q has the form

(5.2)

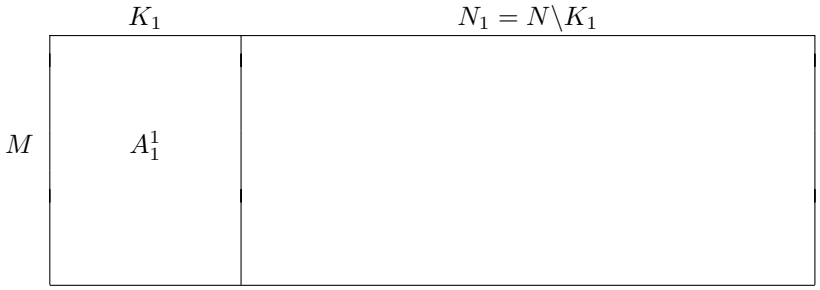


where $K_s \subseteq N$ $(s = 1, \dots, \tau,\ \tau \le m)$, $K_i \bigcap K_j = \phi$ $(i \ne j$ for all $i, j)$, $M = \bigcup_{s=1}^{\tau} R_s$, the submatrices Q_{ss} $(s = 1, \dots, \tau)$ are matrices with orthogonal rows, 0 is a zero submatrix. Note that depending on the selection of K_s the set N_τ can be empty.

To obtain the decomposition (5.1) we consider the process consisting of τ iterations. In each sth iteration we select the set K_s (the number of elements in the set K_s is k_s) and apply the modified partial orthogonalization process with respect to the chosen system of vectors.

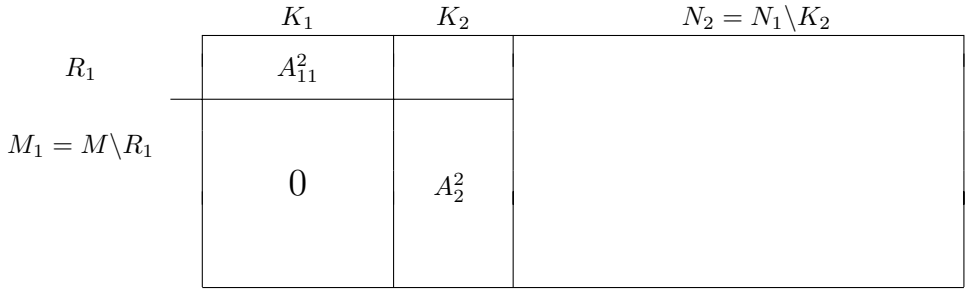
Let us consider this process.

First iteration. Set $A^1 = A$. Select $K_1 \subset N$. Let the rank of the submatrix $A_1^1 \in \mathbb{R}^{m\times k_1}$ be equal to r_1 and the first r_1 rows be linearly independent. Using the modified partial orthogonalization process, after r_1 steps from the matrix A^1, that is,



$A^1 = A$

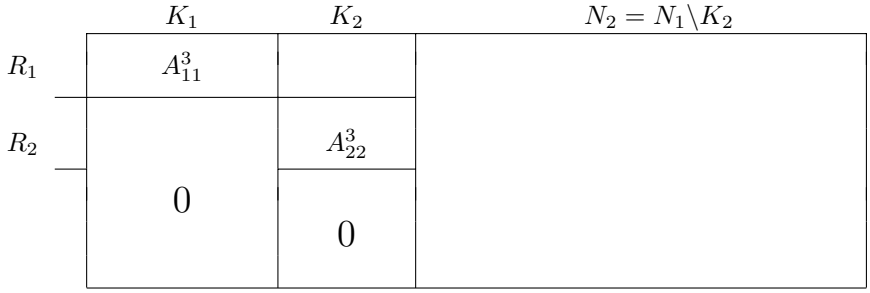
we obtain the matrix A^2 of the form



A^2

Here $R_1 = \{1, \dots, r_1\}$ and $A_{11}^2 \in \mathbb{R}^{r_1\times k_1}$ is a matrix with orthogonal rows. Since in this process the modified orthogonalization goes with respect to the rows of the submatrix $A_1^1 \in \mathbb{R}^{m\times k_1}$, then the rank of A_1^1 is calculated during this process.

Second iteration. Select $K_2 \subset N_1$. Let the rank of the submatrix $A_2^2 \in \mathbb{R}^{(m-r_1)\times k_2}$ be equal to r_2 and the first r_2 rows be linearly independent. Again, using the modified partial orthogonalization process, after r_2 steps from the matrix A^2 we obtain the matrix A^3 of the form



A^3

Here $R_2 = \{r_1 + 1, \ldots, r_1 + r_2\}$ and the submatrix $A^3_{22} \in \mathbb{R}^{r_2 \times k_2}$ is a matrix with orthogonal rows.

Continuing this process, after τ iterations we obtain the matrix of the form (5.2).

Example. Let

$$A^1 = A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & -1 & 1 \\ -1 & -1 & 0 & 0 & 0 & -1 \end{bmatrix}.$$

First iteration. Select $K_1 = \{1, 2\}$. Using the modified partial orthogonalization process we obtain Tables 5.1–5.3.

Second iteration. Select $K_2 = \{3, 4, 5\}$. By performing the second iteration we obtain Table 5.4.

Note that the rows of the diagonal submatrices

$$A^4_{11} = \begin{bmatrix} 1 & 1 \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \quad \text{and} \quad A^4_{22} = \begin{bmatrix} 1 & 1 & -1 \\ \frac{2}{3} & \frac{2}{3} & \frac{4}{3} \end{bmatrix}$$

are orthogonal.

Using this example, by analogy with Theorem 3.1 we will now show how to obtain the decomposition (5.1). Using the columns α_i of Tables 5.1–5.3 we have $A^4 = MA$, where $M = M^3M^2M^1$. From this we obtain

$$A = \widetilde{M} A^4, \tag{5.3}$$

where

$$\widetilde{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & \frac{1}{3} & 1 \end{bmatrix}$$

is the inverse of M.

In order to partially normalize the rows of the matrix A^4 we calculate the norms of the subvectors $\|a^4_1[K_1]\| = \sqrt{2}$, $\|a^4_2[K_1]\| = \frac{\sqrt{2}}{2}$, $\|a^4_3[K_2]\| = \sqrt{3}$, $\|a^4_4[K_2]\| = \frac{2\sqrt{6}}{3}$, and set

$$Q = \begin{array}{c} \begin{matrix} \quad K_1 \quad\quad & \quad\quad & \quad K_2 \quad\quad & \quad\quad & \quad\quad & N_2 \end{matrix} \\ \left[\begin{array}{cc|ccc|c} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ 0 & 0 & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & -\frac{1}{2\sqrt{6}} \end{array}\right] \begin{matrix} R_1 \\ -- \\ R_2 \end{matrix} \end{array},$$

where $Q_{11} \in \mathbb{R}^{r_1 \times k_1}$ and $Q_{22} \in \mathbb{R}^{r_2 \times k_2}$ are the matrices with orthonormal rows.

Further we calculate

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & \frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & \sqrt{3} & 0 \\ 0 & 0 & 0 & \frac{2\sqrt{6}}{3} \end{bmatrix} = \begin{bmatrix} \sqrt{2} & 0 & 0 & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & \sqrt{3} & 0 \\ -\sqrt{2} & 0 & \frac{\sqrt{3}}{3} & \frac{2\sqrt{6}}{3} \end{bmatrix}.$$

TABLE 5.1

K_1		$N_1 = N\backslash K_1$				
1	2	3	4	5	6	α_i
1	1	1	1	1	1	−
1	0	0	0	0	1	$-\frac{1}{2}$
0	0	1	1	−1	1	0
−1	−1	0	0	0	−1	1

$A^1 = A$

TABLE 5.2

	K_1		$N_1 = N\backslash K_1$				
	1	2	3	4	5	6	α_i
	1	1	1	1	1	1	−
R_1	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	−
	0	0	1	1	−1	1	0
	0	0	1	1	1	0	0

A^2

TABLE 5.3
Here $R_1 = \{1, 2\}$.

	K_1		K_2			N_2	
	1	2	3	4	5	6	α_i
	1	1	1	1	1	1	−
R_1	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	−
	0	0	1	1	−1	1	−
	0	0	1	1	1	0	$-\frac{1}{3}$

A^3

TABLE 5.4
Here $R_2 = \{3, 4\}$, $M = R_1 \cup R_2$.

	K_1		K_2			N_2
	1	2	3	4	5	6
	1	1	1	1	1	1
R_1	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$
	0	0	1	1	−1	1
R_2	0	0	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{4}{3}$	$-\frac{1}{3}$

A^4

Due to (5.3) it is easy to see that $A = RQ$. Thus we obtain the following statement.

THEOREM 5.1. *Let $A \in \mathbb{R}^{m\times n}$, $m \le n$, be a full rank matrix. Let $N = \{1, \dots, n\}$, $K = \{K_1, \dots, K_\tau\}$, where*

$$K_s \subseteq N, s = 1, \dots, \tau, \tau \le m, \qquad K_i \bigcap K_j = \phi, i \ne j \text{ for all } i, j, \qquad N_\tau = N\backslash K.$$

Then the matrix A can be represented in the form

$$A = RQ, \tag{5.4}$$

where $R \in \mathbb{R}^{m\times m}$ is a nonsingular lower triangular matrix, and $Q \in \mathbb{R}^{m\times n}$ has the form (5.2)*, where the diagonal submatrices Q_{ss}, $s = 1, \dots, \tau$ have orthogonal rows.*

The decomposition (5.4) is called a *partially orthogonal decomposition*, or a $Q_K R_K$ *decomposition (block approach)*. If all $K_s = K$, then we obtain the decomposition (3.1).

If in the matrix A the number of rows is greater than the number of columns ($m > n$), then

$$A = QR. \tag{5.5}$$

Here $Q \in \mathbb{R}^{m\times n}$ has the form of the transposed matrix with respect to (5.2); that is, Q is a rectangular lower block-diagonal matrix with τ blocks, where each block is a corresponding rectangular matrix with orthogonal columns, and $R \in \mathbb{R}^{n\times n}$ is a nonsingular upper triangular matrix.

6. Calculation of normal pseudosolution by $Q_K R_K$ decomposition (block approach). Again consider the least squares problem $Ax \cong b$, where $A \in \mathbb{R}^{m\times n}$, $m > n$, and $x \in \mathbb{R}^n$ is an unknown column.

By analogy with discourses provided in section 4 using the $Q_K R_K$ decomposition (5.5) we reduce the calculation of the normal pseudosolution to solving the system

$$URx = r,$$

or to solving two systems

$$Uy = r, \tag{6.1}$$

$$Rx = y, \tag{6.2}$$

where $U = Q^T Q$ is a symmetric and "positive definite" matrix[2], $R \in \mathbb{R}^{n\times n}$ is an upper triangular matrix, $r = Q^T b$, and $y \in \mathbb{R}^n$ is an unknown column.

Consider this statement on the example from section 5, setting $A = A^T$, that is,

$$A = \begin{bmatrix} 1 & 1 & 0 & -1 \\ 1 & 0 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 1 & 1 & -1 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ 0 \\ -1 \\ 1 \\ 0 \\ 1 \end{bmatrix}. \tag{6.3}$$

For the matrix A, at $K = \{K_1, K_2\}$, where $K_1 = \{1, 2\}$, $K_2 = \{3, 4, 5\}$,

$$Q = \left[\begin{array}{cc|ccc|c} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ 0 & 0 & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & -\frac{1}{2\sqrt{6}} \end{array}\right]^T \begin{matrix} R_1 \\ -- \\ R_2 \end{matrix} \quad \text{or in the block form,} \quad Q = \begin{bmatrix} Q_{11} & 0 \\ Q_{21} & Q_{22} \\ Q_{31} & Q_{32} \end{bmatrix},$$

$$R = \begin{bmatrix} \sqrt{2} & \frac{\sqrt{2}}{2} & 0 & -\sqrt{2} \\ 0 & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & \sqrt{3} & \frac{\sqrt{3}}{3} \\ 0 & 0 & 0 & \frac{2\sqrt{6}}{3} \end{bmatrix}. \tag{6.4}$$

[2]We will call U *a "positive definite" matrix*, if the diagonal submatrices U_{ss} are positive definite for all s.

Since matrices Q_{11} and Q_{22} have orthonormal columns, $Q_{11}^T Q_{11} = I$, $Q_{22}^T Q_{22} = I$. Therefore,

$$U = Q^T Q = \begin{bmatrix} Q_{11} & 0 \\ Q_{21} & Q_{22} \\ Q_{31} & Q_{32} \end{bmatrix}^T \begin{bmatrix} Q_{11} & 0 \\ Q_{21} & Q_{22} \\ Q_{31} & Q_{32} \end{bmatrix} = \begin{bmatrix} I + D_{11} & D_{12} \\ D_{21} & I + D_{22} \end{bmatrix} = I + D.$$

Here $D_{11} = Q_{21}^T Q_{21} + Q_{31}^T Q_{31}$ is a symmetric matrix; $D_{12} = Q_{21}^T Q_{22} + Q_{31}^T Q_{32}$, $D_{21} = Q_{22}^T Q_{21} + Q_{32}^T Q_{31}$, that is, $D_{12} = D_{21}^T$; $D_{22} = Q_{32}^T Q_{32}$ is a symmetric matrix.

In our example,

$$D_{11} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, \quad D_{12} = \begin{bmatrix} \frac{2}{\sqrt{6}} & \frac{7}{2\sqrt{12}} \\ 0 & -\frac{9}{2\sqrt{12}} \end{bmatrix},$$

$$D_{21} = \begin{bmatrix} \frac{2}{\sqrt{6}} & 0 \\ \frac{7}{2\sqrt{12}} & -\frac{9}{2\sqrt{12}} \end{bmatrix}, \quad D_{22} = \begin{bmatrix} \frac{1}{3} & -\frac{1}{6\sqrt{2}} \\ -\frac{1}{6\sqrt{2}} & \frac{1}{24} \end{bmatrix}, \quad D = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}.$$

Finally,

$$U = I + D = \begin{bmatrix} 1+2 & -1 & \frac{2}{\sqrt{6}} & \frac{7}{2\sqrt{12}} \\ -1 & 1+2 & 0 & -\frac{9}{2\sqrt{12}} \\ \frac{2}{\sqrt{6}} & 0 & 1+\frac{1}{3} & -\frac{1}{6\sqrt{2}} \\ \frac{7}{2\sqrt{12}} & -\frac{9}{2\sqrt{12}} & -\frac{1}{6\sqrt{2}} & 1+\frac{1}{24} \end{bmatrix}. \tag{6.5}$$

The solution of the first system (6.1) with the matrix (6.5) and with the right side $r = Q^T b = (\frac{2}{\sqrt{2}} \quad \frac{2}{\sqrt{2}} \quad \frac{1}{\sqrt{3}} \quad -\frac{1}{2\sqrt{6}})^T$ is $y = (\frac{\sqrt{2}}{2} \quad \frac{\sqrt{2}}{2} \quad 0 \quad 0)^T$, and the solution of the second system (6.2) with the matrix R, which has the form (6.4), is $x = (0 \quad 1 \quad 0 \quad 0)^T$.

Remark 6.1. To calculate the pseudoinverse of A we consider the matrix equation $AX \cong I$. In this case we solve two matrix systems $UY = Q^T$ and $RX = Y$ instead of the systems (4.2) and (4.3), or instead of (6.1) and (6.2) in case of using the decomposition (5.4).

7. On the number of arithmetical operations for solving the least squares problem. The number of operations for reducing the least squares problem to the triangular form (4.3), using the $Q_K R_K$ decomposition (3.7) of the partial orthogonalization process, is equal to

$$\frac{(m+k)n^2}{2} + \frac{(m-k)n^2}{2} + \frac{n^3}{6} = mn^2 + \frac{n^3}{6}, \tag{7.1}$$

where $k \in \{n, n+1, \ldots, m-1\}$; $\frac{(m+k)n^2}{2}$ is the number of operations of the partial orthogonalization process; $\frac{(m-k)n^2}{2}$ is the number of operations for forming the matrix $U \in \mathbb{R}^{n \times n}$; and $\frac{n^3}{6}$ is the number of operations for solving system (4.2) by the Cholesky method.

Let us now consider the block decomposition (5.4) in the case where the number of iterations is equal to 2 and $K_1 = \{1, \ldots, k\}$, $K_2 = \{k+1, \ldots, m\}$. Let the rank

of the submatrix $A_1 \in \mathbb{R}^{k\times n}$ be equal to r. In this case the number of operations for reducing the least squares problem to the triangular form (6.2) is equal to

$$rm(2n-r)+(m-k)(n-r)^2+\frac{n^3}{6}, \tag{7.2}$$

where $n^3/6$ is the number of operations for solving the system (6.1) by the Cholesky method.

Let us now compare the estimations (7.1) and (7.2) with the estimations for the orthogonalization process (mn^2) and for the Householder method ($mn^2-n^3/3$) [11].

In comparison with the orthogonalization process according to (7.1) we have

$$\Delta_1=\left(mn^2+\frac{n^3}{6}\right)-mn^2=\frac{n^3}{6}>0,$$

and in comparison with the Householder method we have

$$\Delta_2=\left(mn^2+\frac{n^3}{6}\right)-\left(mn^2-\frac{n^3}{3}\right)=\frac{n^3}{2}>0.$$

In other words, the number of operations using the partial orthogonalization process is greater than the number of operations using the orthogonalization process by the value $n^3/6$, and it is greater than the number of operations using the Householder method by $n^3/2$. Note that the estimations Δ_1 and Δ_2 do not depend on $k \in \{n, n+1, \ldots, m-1\}$.

For the block case ($\tau = 2$), in comparison with the orthogonalization process according to (7.2) we have

$$\Delta_{k1}=mn^2-\left(rm(2n-r)+(m-k)(n-r)^2+\frac{n^3}{6}\right)=k(n-r)^2-\frac{n^3}{6}, \tag{7.3}$$

and in comparison with the Householder method we have

$$\Delta_{k2}=\left(mn^2-\frac{n^3}{3}\right)-\left(rm(2n-r)+(m-k)(n-r)^2+\frac{n^3}{6}\right)=k(n-r)^2-\frac{n^3}{2}. \tag{7.4}$$

We can see that in the block case (approach) the estimations Δ_{k1} and Δ_{k2} depend on k and r. If, for example, the rank of $A_1 \in \mathbb{R}^{m\times k}$ is equal to $r = n/2$ then according to (7.3) the number of operations is less by Δ_{k1} for any $k > 2n/3$ in comparison with the orthogonalization process, and according to (7.4) it is less by Δ_{k2} for any $k > 2n$ in comparison with the Householder method. Notice that these estimations do not include the number of operations required for pivoting, which is less than mn.

Note also that obtained estimations can be improved by taking into account that the submatrix $U \in \mathbb{R}^{(n-r)\times(n-r)}$ of the symmetric matrix $U \in \mathbb{R}^{n\times n}$ is identity.

8. Conclusions. In many applications, for example, in signal and image processing, it is necessary to identify such a special class of oblique projections for which it is possible to construct fast algorithms [12]. Partial orthogonalization process allows the extraction of such an important subclass of matrix decompositions, that is, to obtain special decompositions of a matrix. Our approach using the modified partial orthogonalization process allows the linear least squares problem to be decomposed into simpler subproblems, yielding computational efficiency. It is shown that the number of operations with our approach (block $Q_K R_K$ decomposition) is less than that of the classical QR decomposition.

Since the proposed method is parametric there are various possibilities to choose the pivot vectors in the sense of their lengths and indices. The choice of the set K will depend on the class of the practical applications.

Acknowledgments. This work was done during the visit of Professor A. Tuniev at Tampere International Center for Signal Processing, Tampere University of Technology in 2000. The authors are grateful to the editor, Professor Moody T. Chu, and the anonymous referees for their valuable comments and helpful suggestions.

REFERENCES

[1] A. Albert, *Regression and the Moore-Penrose Pseudoinverse*, Academic Press, New York, 1972.
[2] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
[3] Å. Björck, *Solving linear least squares problems by Gram-Schmidt orthogonalization*, Nordisk Tidskr. Informations-Behandling, 7 (1967), pp. 1–21.
[4] M. T. Chu, R. E. Funderlic, and G. H. Golub, *A rank-one reduction formula and its applications to matrix factorizations*, SIAM Rev., 37 (1995), pp. 512–530.
[5] G. E. Forsythe and C. B. Moler, *Computer Solution of Linear Algebraic Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1967.
[6] L. V. Foster, *Gaussian elimination with partial pivoting can fail in practice*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1354–1362.
[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.
[8] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 1996.
[9] A. S. Householder, *The Theory of Matrices in Numerical Analysis*, Dover, New York, 1975.
[10] L. Hubert, J. Meulman, and W. Heiser, *Two purposes for matrix factorization: A historical appraisal*, SIAM Rev., 42 (2000), pp. 68–82.
[11] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice–Hall, Englewood Cliffs, NJ, 1974.
[12] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, Prentice–Hall, Upper Saddle River, NJ, 2000.
[13] J. R. Rice, *Experiments on Gram-Schmidt orthogonalization*, Math. Comp., 20 (1966), pp. 325–328.
[14] J. R. Rice, *Matrix Computations and Mathematical Software*, McGraw–Hill, New York, 1981.
[15] G. W. Stewart, *Matrix Algorithms, Vol.* 1*. Basic Decompositions*, SIAM, Philadelphia, 1998.
[16] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*, John Wiley & Sons, New York, 1977.
[17] L. N. Trefethen and R. S. Schreiber, *Average-case stability of Gaussian elimination*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 335–360.
[18] A. D. Tuniev, *Generalization of elimination and complete elimination methods*, Dokl. Akad. Nauk Armyan. SSR Dokl., 71 (1980), pp. 141–146 (in Russian).
[19] A. D. Tuniev, *Pivot vector method and its applications*, Cybernet. Systems Anal., 28 (1992), pp. 99–109.
[20] J. H. Wilkinson, *Error analysis of direct methods of matrix inversion*, J. Assoc. Comput. Mach., 8 (1961), pp. 281–330.
[21] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice–Hall, Englewood Cliffs, NJ, 1963.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 3, pp. 599–620

ON SYMMETRIC EIGENPROBLEMS INDUCED BY THE BIDIAGONAL SVD*

BENEDIKT GROSSER† AND BRUNO LANG†

Abstract. The relatively robust representations (RRR) algorithm is the method of choice to compute highly accurate eigenvector approximations for symmetric tridiagonal matrices. The task of computing singular vector pairs for a bidiagonal matrix $B = U\Sigma V^T$ is closely connected to the RRR algorithm regarding B^TB, BB^T, or the Golub–Kahan matrix T_{GK}. Nevertheless, separate application of the RRR algorithm to these matrices leads to poor results regarding either numerical orthogonality or the residual $\|BV - U\Sigma\|$. It turns out that the coupling strategy proposed in [B. Grosser and B. Lang, *Linear Algebra Appl.*, 358 (2003), pp. 45–70] resolves this problem. This article provides the corresponding perturbation theory: We compare the eigenvalues of the separate and coupled decompositions and explain why singular vector pairs approximated via couplings are of superior quality.

Key words. bidiagonal SVD, relatively robust representations, coupling relations, stability

AMS subject classifications. 15A18, 65G50, 65F15

DOI. 10.1137/S0895479801394829

1. Introduction.

The singular value decomposition (SVD) of a real $n \times n$ upper bidiagonal matrix is given by $B = U\Sigma V^T$ (bidiagonal SVD, **bSVD**). The diagonal matrix $\Sigma = \mathtt{diag}([\sigma_1, \ldots, \sigma_n])$ contains the singular values in descending order, while the left and right singular vectors make up the orthogonal matrices U and V. It is possible to reduce any general complex matrix to real bidiagonal form by a sequence of unitary transformations [17].

We can establish connections to the tridiagonal symmetric eigenproblem (**tSEP**) via the normal equations $\hat{T} = B^TB = V\Sigma^2V^T$ and $\check{T} = BB^T = U\Sigma^2U^T$ or the so-called *Golub–Kahan matrix* T_{GK}. The latter is obtained by applying the "perfect shuffle" permutation [22] to the Jordan–Wielandt form of B: $T_{\mathrm{GK}} = P_{ps}\left[\begin{smallmatrix} 0 & B \\ B^T & 0 \end{smallmatrix}\right]P_{ps}^T$. This permutation interleaves the diagonal and superdiagonal elements of B onto the subdiagonal of the $2n \times 2n$ symmetric tridiagonal matrix T_{GK}, whereas its diagonal is zero. An eigendecomposition $T_{\mathrm{GK}} = \tilde{X} \cdot \tilde{\Lambda} \cdot \tilde{X}^T$ can be related to the **bSVD** of B, noting that the eigenvalues are given by $\{\sigma_j \mid j = 1 : n\} \cup \{-\sigma_j \mid j = 1 : n\}$. In addition we can extract the (scaled) left and right singular vectors of B from the even and the odd rows of $\tilde{X}$.

The tight relations between the **bSVD** and the **tSEP** immediately suggest four methods for computing the SVD of a bidiagonal matrix B:

1. Compute the eigendecomposition $\hat{T} = \hat{X}\hat{\Lambda}\hat{X}^T$ of the symmetric tridiagonal matrix $\hat{T} = B^TB$. The eigenvalues $\hat{\lambda}_j = \sigma_j^2$ give the singular values σ_j of B, and the eigenvectors $\hat{x}_j = v_j$ are B's right singular vectors. To obtain the left singular vectors u_j, compute $u_j = \frac{1}{\sigma_j}Bv_j$.
2. Determine the singular values $\sigma_j = \check{\lambda}_j^{1/2}$ and the left singular vectors $u_j = \check{x}_j$

*Received by the editors, September 6, 2001; accepted for publication (in revised form) by M. Chu May 7, 2004; published electronically March 3, 2005.
http://www.siam.org/journals/simax/26-3/39482.html

†Fachbereich Mathematik und Naturwissenschaften, Bergische Universität Wuppertal, Gaußstr. 20, D-42097 Wuppertal, Germany (grosser@math.uni-wuppertal.de, lang@math.uni-wuppertal.de).
This work was partially funded by Deutsche Forschungsgemeinschaft, Geschäftszeichen Fr 755/6-1 and Fr 755/6-2.

from the eigendecomposition $\check{T} = \check{X}\check{\Lambda}\check{X}^T$ of $\check{T} = BB^T$ and the right singular vectors v_j by solving the linear systems $Bv_j = \sigma_j u_j$.

3. Compute both eigendecompositions $\hat{T} = \hat{X}\hat{\Lambda}\hat{X}^T$ and $\check{T} = \check{X}\check{\Lambda}\check{X}^T$ to obtain the singular values $\sigma_j = \hat{\lambda}_j^{1/2} = \check{\lambda}_j^{1/2}$ and the left and right singular vectors $u_j = \check{x}_j$ and $v_j = \hat{x}_j$, respectively.
4. Compute the eigendecomposition $T_{\mathrm{GK}} = \tilde{X}\tilde{\Lambda}\tilde{X}^T$. The nonnegative eigenvalues $\tilde{\lambda}_j$ are B's singular values, and the corresponding eigenvectors $\tilde{x}_j$ yield the singular vectors v_j and u_j by extracting the odd-numbered (even-numbered, resp.) components from $\tilde{x}_j$.

Each of these methods involves computing a full eigendecomposition $T = X\Lambda X^T$ of an $n \times n$ or $2n \times 2n$ symmetric tridiagonal matrix T. A few years ago, the relatively robust representations (RRR) algorithm [10] was discovered, which is able to determine such an eigendecomposition with $\mathcal{O}(n^2)$ operations such that

- $\|X^T X - I\|$ is small (i.e., the computed eigenvectors are numerically orthogonal), and
- the residuals $\|Tx_j - x_j\lambda_j\|$ are small for all j (i.e., the computed eigenvalues and eigenvectors are "consistent").

Note that $\hat{T} = B^T B$, $\check{T} = BB^T$, and T_{GK} *are* relatively robust representations, and therefore the RRR algorithm encounters no problems in computing the respective eigendecompositions. (For a thorough discussion of the existence of (partial) RRRs, see [9, 10].)

Despite this fact, unfortunately, none of the above four methods is able to produce a "consistent" SVD of the matrix B, in the sense that

- both $\|U^T U - I\|$ and $\|V^T V - I\|$ are small, i.e., the computed singular vectors u_j and v_j are numerically orthogonal, *and*
- the residuals $\|Bv_j - u_j\sigma_j\|$ are small.

For methods 1 and 2 it is well known that the multiplication with B (the linear solves) may destroy the orthogonality of U (V, resp.).

The reason for the failure of the third approach is more subtle. The RRR algorithm *does* produce consistent eigendecompositions $\hat{T} = \hat{X}\hat{\Lambda}\hat{X}^T$ and $\check{T} = \check{X}\check{\Lambda}\check{X}^T$. More precisely, for each cluster of very close eigenvalues $\lambda_k \approx \lambda_{k+1} \approx \cdots \approx \lambda_{k+\ell}$, a numerically orthogonal basis of the corresponding invariant subspace is computed. But *there is no guarantee that the computed eigenvectors are close to the exact eigenvectors.* In fact, the orientation of the computed basis is almost random, depending heavily on the rounding errors made during the RRR algorithm, and the RRR algorithm owes much of its immense success to the fact that *any* orthogonal basis of the invariant subspace leads to small residuals. However, this is no longer true in the **bSVD** context, since the left and right singular vectors are *coupled* via the relations $Bv_j = u_j\sigma_j$. That is, if we apply the RRR algorithm *independently* to $\hat{T}$ and $\check{T}$, and if σ_j belongs to a cluster, then v_j and u_j will be (almost) random unit vectors from $\hat{T}$'s and $\check{T}$'s invariant subspaces corresponding to the cluster, and $\|Bv_j - u_j\sigma_j\|$ may be large. In section 4.3 we will illustrate this fact with a numerical example and give a geometric interpretation. This interpretation also shows that large residuals can be avoided if the eigenvalues $\hat{\lambda}_j$ and $\check{\lambda}_j$ of certain *shifted* matrices $\hat{T} - \bar{\mu}^2 I$ and $\check{T} - \bar{\mu}^2 I$ agree to high *relative* accuracy. In section 4.2 we will show that applying the RRR algorithm independently to $\hat{T}$ and $\check{T}$ cannot guarantee small relative differences of these eigenvalues.

The failure of the fourth approach can be explained in a similar manner. While extracting U and V from the *exact* eigenvectors $\tilde{x}_j$ would produce orthogonal matrices,

the computed U and V may be far from orthogonality if an arbitrary orthonormal base $\tilde{x}_k, \ldots, \tilde{x}_{k+\ell}$ of an invariant subspace corresponding to a cluster of eigenvalues is used instead.

To summarize, the failure of the standard RRR algorithm (and any other **tSEP** algorithm as well) used in a "black-box" manner in the **bSVD** context cannot be attributed to deficiencies of these algorithms themselves, but to the fact that—by their very design—they do not take into account the coupling of the left and right singular vectors.

In [18, 19] we have proposed an alternative approach: Run the RRR algorithm explicitly on one of the tridiagonal matrices, say, T_{GK}. This will produce certain intermediate quantities, such as decompositions $\tilde{T} - \bar{\mu} I = \tilde{L}\tilde{D}\tilde{L}^T$ of shifted matrices. The corresponding quantities $\hat{L}$, $\hat{D}$ and $\check{L}$, $\check{D}$ for the matrices $\hat{T}$ and $\check{T}$ can be obtained directly from $\tilde{L}$, $\tilde{D}$ via so-called *coupling relations*, so that we can apply the RRR algorithm implicitly to $\hat{T}$ and $\check{T}$ without actually running it. In section 5 we will show that in this way the respective eigenvalues $\hat{\lambda}_j$ and $\check{\lambda}_j$ do agree to high relative accuracy, which in turn leads to small residuals. In section 5.4 we will briefly outline how to incorporate the coupling relations into the RRR algorithm and in particular we will discuss the question for which of the matrices $\hat{T} = B^T B$, $\check{T} = BB^T$, or T_{GK} the RRR algorithm should be run explicitly. The resulting RRR algorithm for the **bSVD** is described in [19]. The error analyses given in the present paper provide the theoretical justification for this approach.

2. The tridiagonal RRR algorithm in the context of the bSVD. The RRR algorithm computes a full eigenvector basis of a symmetric tridiagonal matrix T with a complexity of $\mathcal{O}(n^2)$. It can be implemented in a fast and accurate way and is inherently parallel. Although we give a very brief overview of the RRR algorithm in the following and a geometric interpretation in section 4.3, we recommend that the reader has some acquaintance with the relevant work in [8, 9, 10].

Starting with a symmetric tridiagonal matrix T, the RRR algorithm computes eigenvectors for isolated eigenvalues using so-called "twisted factorizations," $T - \bar{\lambda} I = N_k G_k N_k^T$, where $\bar{\lambda}$ is a very good approximation to an eigenvalue λ_j of T [7, 11, 12, 15, 16]. Twisted factorizations are a generalization of the standard symmetric indefinite factorizations $T - \bar{\lambda} I = LDL^T$. Eigenvalue clusters are treated by applying this technique *recursively* to $T - \alpha I = LDL^T$, where a different shift α is chosen for each cluster in order to increase the relative distances of the eigenvalues $\lambda_j - \alpha$ within the cluster. (The relative distance of two numbers, p and q, is given by $|p - q|/|p|$.) Therefore the RRR algorithm can use the original data T only on the first level. By contrast, we have to operate with preprocessed matrices at deeper recursion levels.

While the classical RRR algorithm for the **tSEP** is applied only to a single base matrix (the symmetric tridiagonal matrix T given by its entries), we have three choices—$B^T B$, BB^T, and T_{GK}—when computing singular vector pairs for the **bSVD**; see Figure 2.1.

Applying the RRR algorithm to these tridiagonal matrices requires considering factorizations

$$B^T B - \bar{\mu}^2 I = \hat{L}\hat{D}\hat{L}^T, \quad BB^T - \bar{\mu}^2 I = \check{L}\check{D}\check{L}^T, \quad \text{and} \quad T_{\mathrm{GK}} - \bar{\mu} I = \tilde{L}\tilde{D}\tilde{L}^T, \tag{2.1}$$

where the shift parameter $\bar{\mu}$ is chosen close to a singular value of B, i.e., as a floating point number approximating a certain σ_j. Throughout this article, we use the $\hat{}$, $\check{}$, and $\tilde{}$ superscripts to distinguish between the decompositions belonging to $B^T B$, BB^T, and T_{GK}, respectively.

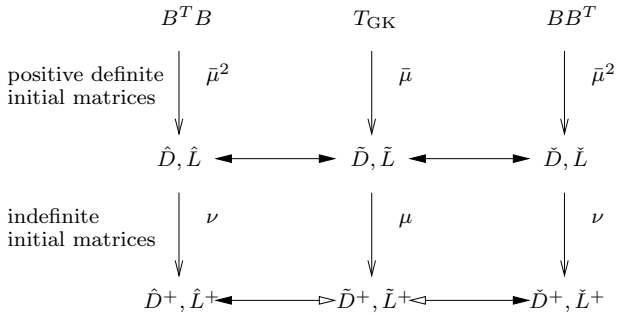


FIG. 2.1. *Forming LDL^T factorizations for translates of the normal equations B^TB and BB^T and of the Golub–Kahan matrix T_{GK}. The shift parameters μ, ν for the indefinite initial matrices are chosen such that $(\bar{\mu}+\mu)^2 = \bar{\mu}^2 + \nu$. Setting up the respective decompositions can be done either by explicit factorizations (vertical arrows) or by implicit couplings (horizontal arrows).*

Clusters of singular values require recursion in the RRR algorithm and therefore make it necessary to consider decompositions

$$\hat{L}\hat{D}\hat{L}^T - \nu I = \hat{L}^+\hat{D}^+(\hat{L}^+)^T, \tag{2.2}$$

$$\check{L}\check{D}\check{L}^T - \nu I = \check{L}^+\check{D}^+(\check{L}^+)^T, \tag{2.3}$$

$$\tilde{L}\tilde{D}\tilde{L}^T - \mu I = \tilde{L}^+\tilde{D}^+(\tilde{L}^+)^T \tag{2.4}$$

as well, where the *initial matrices* $\hat{L}\hat{D}\hat{L}^T$, etc., on the left-hand side are formed according to (2.1). Thus they are typically *indefinite*. By contrast, the three factorizations from (2.1) are referred to as the *case of positive definite initial matrices*; see Figure 2.1. As in (2.2)–(2.4), the + label will be used in the remainder of the article as a symbol for successive factorizations at deeper recursion levels. We point out that this notation is different from [8, 9, 10]. There the + and − labels are used in the context of the twisted factorizations ("from top to bottom" versus "from bottom to top").

In exact arithmetic, the eigenvalues (again in descending order) of the right-hand sides of (2.1) are related to the singular values of B via $\hat{\lambda}_j = \check{\lambda}_j = \sigma_j^2 - \bar{\mu}^2$ and $\tilde{\lambda}_j = \sigma_j - \bar{\mu}$, $\tilde{\lambda}_{2n-j+1} = -\sigma_j - \bar{\mu}$. Analogously, choosing the shift parameters in (2.2)–(2.4) such that $(\bar{\mu}+\mu)^2 = \bar{\mu}^2 + \nu$ leads to the eigenvalues of the right-hand sides being given by $\hat{\lambda}_j^+ = \check{\lambda}_j^+ = \sigma_j^2 - (\bar{\mu}+\mu)^2$ and $\tilde{\lambda}_j^+ = \sigma_j - (\bar{\mu}+\mu)$.

Since there are inevitable rounding errors when computing the corresponding decompositions in floating point arithmetic, the above relations do not hold in practice. For example, we have $\hat{\lambda}_j^+ \neq \check{\lambda}_j^+$. In section 4.3 we show that $\hat{\lambda}_j^+$ and $\check{\lambda}_j^+$ must be very close in order to obtain approximations to singular vectors u_j and v_j with a small residual $\|Bv_j - \sigma_j u_j\|$. Note that we need not distinguish between the *computed* and *exact* eigenvalues of the computed decompositions since we employ the concept of relatively robust representations defined in section 4.1 to ensure that these eigenvalues can be approximated to high relative accuracy. Therefore the focus of this article is on *exact eigenvalues of perturbed factorizations*. In particular we will show that the quality of the singular vector pairs depends heavily on the way the decompositions are obtained.

There are several ways to determine factorized representations of the three base matrices: If we compute the decompositions separately, i.e., independently from each other (vertical arrows in Figure 2.1), it turns out that the exact eigenvalues of the

TABLE 2.1

Deviation of eigenvalues corresponding to normal equations. For separate factorizations we have absolute *bounds on the eigenvalues' deviation. If we use couplings, then the respective eigenvalues are close with small* relative *deviations.*

	Separate	Coupling	
positive definite initial matrices	$\left\lvert\hat{\lambda}_j - \check{\lambda}_j\right\rvert = \mathcal{O}(\epsilon\bar{\mu}^2)$	$\frac{\lvert\hat{\lambda}_j - \check{\lambda}_j\rvert}{\lvert\hat{\lambda}_j\rvert} = \mathcal{O}(\epsilon)$	
indefinite initial matrices	$\left\lvert\hat{\lambda}_j^+ - \check{\lambda}_j^+\right\rvert = \mathcal{O}(\epsilon\mu^2)$	$\tilde{D}^+ \to [\hat{D}^+, \check{D}^+]$	$\hat{D}^+ \to \tilde{D}^+ \to \check{D}^+$
		$\frac{\lvert\hat{\lambda}_j^+ - \check{\lambda}_j^+\rvert}{\lvert\hat{\lambda}_j^+\rvert} = \mathcal{O}(\epsilon)$	only a posteriori and numerically

perturbed factorizations differ significantly. This effects poor results regarding either numerical orthogonality or the residual $\|BV - U\Sigma\|$. The alternative is to use coupling transformations relating the data of the respective LDL^T decompositions implicitly (horizontal arrows in Figure 2.1). Determining numerical data with the help of couplings leads to favorable results: We show that the respective eigenvalues agree to most of their digits, independently from their magnitude. This leads to superior quality of the approximated singular vector pairs.

In Table 2.1 we summarize the main results of this article. They are derived as follows: First we review appropriate factorization procedures as well as a corresponding mixed stability analysis in section 3. In section 4 we first introduce the concept of relatively robust representations and then show why eigenvalues of separate factorizations generally do have unfavorable *absolute* deviations; see also the second column in Table 2.1. (Since $\bar{\mu}$ is typically chosen to approximate σ_j, we have $\hat{\lambda}_j = \check{\lambda}_j = \mathcal{O}(\epsilon\bar{\mu}^2)$, where ϵ is the machine precision. Thus $\hat{\lambda}_j$ and $\check{\lambda}_j$ may differ in almost all significant digits.) As a solution we propose to factorize only one of the three base matrices explicitly, while the remaining decompositions are set up implicitly. To this aim we present a set of coupling transformations in section 5 and prove that the eigenvalues of these coupled factorizations agree to high *relative* accuracy; see the right column in Table 2.1. Concluding remarks briefly sketch how these results can be incorporated into an extension of the RRR algorithm to the **bSVD**.

Most of the matrices we consider contain only a few nonzero entries. For a given n-vector x we define $\texttt{diag}(x, k)$ as a square matrix of order $n + |k|$ with the elements of x on the kth diagonal. Thus, a lower unit bidiagonal matrix can be represented by $L = I + \texttt{diag}([l_1, \ldots, l_{n-1}], -1)$, where I is the identity matrix of dimension n. Diagonal matrices are described with $D = \texttt{diag}([d_1, \ldots, d_n])$. Sticking to this notation, we also describe certain auxiliary quantities as diagonal matrices S and P. The upper bidiagonal matrix B is given by its entries $B = \texttt{diag}([a_1, \ldots, a_n]) + \texttt{diag}([b_1, \ldots, b_{n-1}], 1)$. Using the auxiliary vector $c := [a_1, b_1, a_2, \ldots, b_{n-1}, a_n]$, the Golub–Kahan matrix can be written as $T_{\mathrm{GK}} = \texttt{diag}(c, 1) + \texttt{diag}(c, -1)$. Also recall that the $\hat{}$, $\check{}$, and $\tilde{}$ superscripts refer to decompositions belonging to B^TB, BB^T, and T_{GK}, respectively, and that the $+$ label indicates right-hand sides of factorizations (2.2)–(2.4) at deeper recursion levels.

3. Factorizations. In this section we review certain procedures to compute LDL^T factorizations. We demonstrate that these so-called *differential quotient-difference transformations* **(qd)** are well suited for a relative mixed stability analysis.

3.1. Building decompositions with qd-like recurrences. First we point out that explicitly forming B^TB or BB^T typically causes a substantial loss of accuracy of the singular values and is no longer considered.

The classical factorization procedure for $B^TB - \bar{\mu}^2 I = \hat{L}\hat{D}\hat{L}^T$ is based on equating the diagonal and off-diagonal elements

$$a_{i+1}^2 + b_i^2 - \bar{\mu}^2 = \hat{d}_{i+1} + \hat{d}_i \hat{l}_i^2 \qquad \text{and} \qquad a_i b_i = \hat{d}_i \hat{l}_i.$$

Thus we can determine the elements $\hat{d}_i$ and $\hat{l}_i$ alternatingly [21]. By introducing the auxiliary variable

$$\hat{s}_i = \hat{d}_i - a_i^2 = \frac{b_{i-1}\hat{l}_{i-1}}{a_{i-1}}\hat{s}_{i-1} - \bar{\mu}^2$$

it is possible to avoid cancellation which might occur in the classical version. The usage of the auxiliary variables $\hat{S}$ leads to the so-called differential stationary **qd** transformation described in Algorithm 1. For computational purposes it is convenient to define $\hat{q}_i = a_i^2$ and $\hat{e}_i = a_i b_i$ and to aggregate these numbers in a vector $\hat{Z}$. Note that these quantities are independent from the shift parameter $\bar{\mu}$ and can be computed in advance.

ALGORITHM 1. Factorize $B^TB - \bar{\mu}^2 I$ (left) and $BB^T - \bar{\mu}^2 I$ (right).

Input: $\hat{Z}, \bar{\mu}$	**Input:** $\check{Z}, \bar{\mu}$
Output: $\hat{D}, \hat{L}, \hat{S}$	**Output:** $\check{D}, \check{L}, \check{P}$
$\hat{s}_1 = -\bar{\mu}^2$	$\check{p}_1 = a_1^2 - \bar{\mu}^2$
for $i = 1 : n-1$ **do**	**for** $i = 1 : n-1$ **do**
$\hat{d}_i = \hat{s}_i + \hat{q}_i$	$\check{d}_i = \check{p}_i + \check{q}_i$
$\hat{l}_i = \frac{\hat{e}_i}{\hat{d}_i}$	$\check{l}_i = \frac{\check{e}_i}{\check{d}_i}$
$\hat{s}_{i+1} = \frac{\hat{e}_i \hat{l}_i}{\hat{q}_i}\hat{s}_i - \bar{\mu}^2$	$\check{p}_{i+1} = \frac{\check{e}_i \check{l}_i}{\check{q}_i}\check{p}_i - \bar{\mu}^2$
end for	**end for**
$\hat{d}_n = \hat{s}_n + \hat{q}_n$	$\check{d}_n = \check{p}_n$

Analogously, comparing entries in $BB^T - \bar{\mu}^2 I = \check{L}\check{D}\check{L}^T$ yields

$$a_{i+1}^2 + b_{i+1}^2 - \bar{\mu}^2 = \check{d}_{i+1} + \check{d}_i \check{l}_i^2 \qquad \text{and} \qquad a_{i+1} b_i = \check{d}_i \check{l}_i,$$

leading to the auxiliary quantities

$$\check{p}_i = \check{d}_i - b_i^2 = \frac{a_i \check{l}_{i-1}}{b_{i-1}}\check{p}_{i-1} - \bar{\mu}^2.$$

The resulting differential progressive **qd** transformation is also given in Algorithm 1. The vector $\check{Z}$ contains the predefined variables $\check{q}_i = b_i^2$ and $\check{e}_i = a_{i+1} b_i$.

For factorizing $T_{\mathrm{GK}} - \bar{\mu} I = \tilde{L}\tilde{D}\tilde{L}^T$ we exploit the fact that the diagonal entries of the Golub–Kahan matrix are zero to obtain

$$-\bar{\mu} = \tilde{d}_{i+1} + \tilde{d}_i \tilde{l}_i^2 \qquad \text{and} \qquad c_i = \tilde{d}_i \tilde{l}_i.$$

Here, the resulting Algorithm 2 does not involve auxiliary variables.

ALGORITHM 2. Factorize $T_{\mathrm{GK}} - \bar{\mu} I = \tilde{L}\tilde{D}\tilde{L}^T$.

Input: $c := [a_1, b_1, a_2, \ldots, b_{n-1}, a_n]$, $\bar{\mu}$
Output: $\tilde{D}, \tilde{L}$

1: $\tilde{d}_1 = -\bar{\mu}$
2: **for** $i = 1 : 2n-1$ **do**
3: $\quad \tilde{l}_i = \frac{c_i}{\tilde{d}_i}$
4: $\quad \tilde{d}_{i+1} = -\frac{c_i^2}{\tilde{d}_i} - \bar{\mu} = -c_i \tilde{l}_i - \bar{\mu}$
5: **end for**

Finally, the indefinite cases $LDL^T - \tau I = L^+ D^+ (L^+)^T$ from (2.2)–(2.4) give the relations

$$d_{i+1} + d_i l_i^2 - \tau = d_{i+1}^+ + d_i^+ (l_i^+)^2 \qquad \text{and} \qquad d_i l_i = d_i^+ l_i^+ .$$

Introducing the quantities $s_i = d_i^+ - d_i$ yields the (differential stationary) **qd** transformation in Algorithm 3.

ALGORITHM 3. Factorize $LDL^T - \tau I = L^+ D^+ (L^+)^T$.

Input: D, L, τ
Output: D^+, L^+, S

1: $s_1 = -\tau$
2: **for** $i = 1 : n-1$ **do**
3: $\quad d_i^+ = d_i + s_i$
4: $\quad l_i^+ = \frac{d_i l_i}{d_i^+}$
5: $\quad s_{i+1} = l_i^+ l_i s_i - \tau$
6: **end for**
7: $d_n^+ = d_n + s_n$

All these transformations turn out to have very pleasant features with respect to roundoff error analysis [8, 10, 14, 21]. Moreover, in the case of positive definite initial matrices we can make use of the auxiliary variables $\hat{S}$ and $\check{P}$ to establish coupling formulas for converting one of the factorizations into the others [13, 19].

3.2. Mixed stability analysis. In the remainder of this section we recapitulate the fundamental ideas and techniques of mixed stability analysis. We consider this useful for a better understanding of the results presented in Theorems 5.2 and 5.4. The experienced reader might go directly to the next section.

In the following we discuss the numerical quality of the data generated by Algorithms 1, 2, and 3 in floating point arithmetic. To this aim we use the standard model for elementary floating point operations $\circ \in \{+, -, *, /\}$,

$$\mathrm{fl}(x \circ y) = (x \circ y)(1 + \epsilon_1) = (x \circ y)/(1 + \epsilon_2). \tag{3.1}$$

The relative error terms ϵ_1 and ϵ_2 depend on the operation $\circ$, its arguments x and y, and the underlying arithmetic. Their absolute values are bounded by the machine precision ϵ.

We now study the roundoff errors introduced when determining the factorization $B^T B - \bar{\mu}^2 I = \hat{L}\hat{D}\hat{L}^T$ with the left-hand side of Algorithm 1. We point out that we

interpret $\hat{Z}$ as input data for the computation. Note that the preprocessed quantities $\hat{q}_i$ and $\hat{e}_i$ are the result of a floating point operation themselves:

$$\hat{q}_i = \mathrm{fl}(a_i^2) = a_i^2(1+\epsilon_{a_i^2}) = \hat{a}_i^2 \quad \text{with} \quad \hat{a}_i = a_i\sqrt{1+\epsilon_{a_i^2}},$$
$$\hat{e}_i = \mathrm{fl}(a_i b_i) = a_i b_i(1+\epsilon_{a_i b_i}) = \hat{a}_i\hat{b}_i \quad \text{with} \quad \hat{b}_i = b_i(1+\epsilon_{a_i b_i})/\sqrt{1+\epsilon_{a_i^2}}.$$

Thus performing Algorithm 1 with $\hat{Z}$ means working on a slightly perturbed bidiagonal matrix $\hat{B}$. We will discuss the consequences of these small relative componentwise perturbations in section 4.1.

Following the basic idea of a *mixed stability analysis*, we first impose small relative perturbations on the input data $\hat{Z}$. We then apply Algorithm 1 to the resulting quantities $\widehat{Z}$ in *exact* arithmetic to get $\widehat{D}, \widehat{L}$, and $\widehat{S}$. (Thus the $\widehat{\ }$ superscript denotes perturbed $\hat{\ }$ quantities.) In the last step we interpret these results as componentwise perturbations of the *computed* output data $\hat{D}, \hat{L}$, and $\hat{S}$. A remarkable point of this approach is that it is almost independent from the shift parameter $\bar{\mu}$. The only requirement is the absence of over- and underflows.

Note that the following lemma is very similar to [14, Theorem 4] and [8, Theorem 4]. We prove it nevertheless to give the reader an idea of the underlying concepts.

LEMMA 3.1 (mixed stability analysis for $B^TB - \bar{\mu}^2 I = \hat{L}\hat{D}\hat{L}^T$). *Suppose that the left-hand side of Algorithm* 1 *can be performed without over- and underflow. Then the following diagram commutes:*

$$\begin{array}{ccc} \hat{Z} & \xrightarrow[\text{Alg. 1, left}]{\text{computed}} & \hat{D}, \hat{L}, \hat{S} \\ \Big\downarrow \text{perturbation (3.2)} & & \Big\uparrow \text{perturbation (3.3)} \\ \widehat{Z} & \xrightarrow[\text{Alg. 1, left}]{\text{exact}} & \widehat{D}, \widehat{L}, \widehat{S} \end{array}$$

The multiplicative perturbations are given as follows:

$$\left.\begin{array}{rcl} \widehat{q}_i &=& \hat{q}_i\left(1+\kappa_q^{(i)}\epsilon\right), \quad \left|\kappa_q^{(i)}\right| \leq 1 \\ \widehat{e}_i &=& \hat{e}_i\left(1+\kappa_e^{(i)}\epsilon\right), \quad \left|\kappa_e^{(i)}\right| \leq 3+\mathcal{O}(\epsilon) \end{array}\right\}, \tag{3.2}$$

$$\left.\begin{array}{rcl} \widehat{d}_i &=& \hat{d}_i\left(1+\kappa_d^{(i)}\epsilon\right), \quad \left|\kappa_d^{(i)}\right| \leq 2+\mathcal{O}(\epsilon) \\ \widehat{l}_i &=& \hat{l}_i\left(1+\kappa_l^{(i)}\epsilon\right), \quad \left|\kappa_l^{(i)}\right| \leq 3+\mathcal{O}(\epsilon) \\ \widehat{s}_i &=& s_i\left(1+\kappa_s^{(i)}\epsilon\right), \quad \left|\kappa_s^{(i)}\right| \leq 1 \end{array}\right\}. \tag{3.3}$$

Proof. We first consider the computations in exact arithmetic:

$$\begin{array}{rcll} \widehat{d}_i &=& \widehat{s}_i + \widehat{q}_i & (A1), \\ \widehat{l}_i &=& \frac{\widehat{e}_i}{\widehat{s}_i+\widehat{q}_i} & (B1), \\ \widehat{s}_{i+1} &=& \frac{\widehat{e}_i\widehat{l}_i}{\widehat{q}_i}\widehat{s}_i - \bar{\mu}^2 & (C1). \end{array}$$

We use the model of floating point operations (3.1) to describe the computation in finite precision. Thus

$$\begin{aligned} \hat{d}_i &= (\hat{s}_i + \hat{q}_i)/(1+\epsilon_+^{(i)}) \qquad (A2), \\ \hat{l}_i &= \tfrac{\hat{e}_i}{\hat{s}_i+\hat{q}_i}(1+\epsilon_+^{(i)})(1+\epsilon_/^{(i)}) \quad (B2). \end{aligned}$$

For the sake of simplicity we drop the $^{(i)}$ superscripts whenever the correlation is clear from the context. Proceeding like this we obtain

$$\hat{s}_{i+1} = \left(\frac{\hat{e}_i \hat{l}_i}{\hat{q}_i} \hat{s}_i (1+\epsilon_*)(1+\epsilon_{**})(1+\epsilon_{//}) - \bar{\mu}^2 \right) / (1+\epsilon_{\bar{\mu}^2}^{(i+1)}),$$

which can be written as

$$(1+\epsilon_{\bar{\mu}^2}^{(i+1)})\hat{s}_{i+1} = \tfrac{\hat{e}_i \hat{l}_i}{\hat{q}_i}\hat{s}_i(1+\delta') - \bar{\mu}^2 \quad (C2),$$

using $1+\delta' = (1+\epsilon_*)(1+\epsilon_{**})(1+\epsilon_{//})$. We can omit the respective superscripts except for the ones in $\epsilon_{\bar{\mu}^2}$.

Now there is some freedom in describing the perturbations $\hat{Z} \longrightarrow \widehat{Z}$ and $[\hat{D}, \hat{L}, \hat{S}] \longrightarrow [\hat{D}, \hat{L}, \hat{S}]$. For example, if we choose

$$\begin{aligned} \widehat{q}_i &= \hat{q}_i \left(1+\epsilon_{\bar{\mu}^2}^{(i)}\right), \\ \widehat{e}_i &= \hat{e}_i \sqrt{(1+\delta')(1+\delta'')}, \\ \widehat{s}_i &= \hat{s}_i \left(1+\epsilon_{\bar{\mu}^2}^{(i)}\right), \\ \widehat{d}_i &= \hat{d}_i \left(1+\epsilon_{\bar{\mu}^2}^{(i)}\right)(1+\epsilon_+), \\ \widehat{l}_i &= \hat{l}_i \sqrt{(1+\delta')/(1+\delta'')} \end{aligned}$$

with $1+\delta'' = (1+\epsilon_{\bar{\mu}^2}^{(i)})(1+\epsilon_+)(1+\epsilon_/)$, it is easy to prove $(A1) \Leftrightarrow (A2)$, $(B1) \Leftrightarrow (B2)$, and $(C1) \Leftrightarrow (C2)$. ∎

The main advantage of mixing forward and backward stability analysis for this kind of transformation is that we need only to impose small *relative* perturbations on single components of the decompositions. Within the context of RRR (cf. section 4.1) this means that we are able to guarantee high relative accuracy of the corresponding eigenvalues.

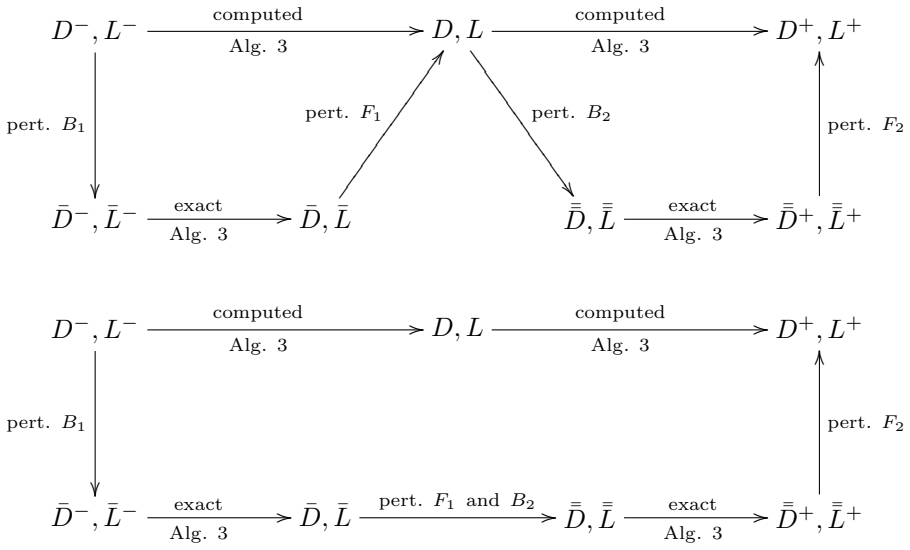
REMARK 3.2 (successive factorizations). *A similar mixed stability analysis for the factorization*

$$LDL^T - \tau I = L^+ D^+ (L^+)^T$$

generated with Algorithm 3 *is given in* [10]. *It is obvious that several transformations computed one after another can be described with small relative errors. In particular, in combination with a preceding factorization*

$$L^- D^- (L^-)^T - \tau^- I = LDL^T$$

the following diagrams are equivalent:



4. Exact eigenvalues of perturbed factorizations. Given the RRR algorithm for the **tSEP**, a natural approach for determining the **bSVD** would be to apply it as a black box to solve the normal equations $B^TB = V\Sigma^2V^T$ and $BB^T = U\Sigma^2U^T$ separately. But in this section it is shown that the computed approximations to U and V can be poorly coupled for clusters of large singular values; i.e., the residual $\|B\bar{v} - \bar{\sigma}\bar{u}\|$ strongly exceeds acceptable thresholds. An explanation for this phenomenon is given in Theorem 4.4 together with a geometric interpretation in section 4.3. It is prepared by reviewing the concept of RRRs.

4.1. Relatively robust representations. In the previous section we demonstrated that differential **qd** transformations are well suited for a mixed stability analysis by introducing small relative perturbations into the components. If in addition a matrix shares the property that it can be written as an RRR (see below), then we can even conclude small relative changes in the eigenvalues.

DEFINITION 4.1 (relatively robust representations). *An* RRR *of a matrix A is a set of numbers $\mathcal{A} = \{\alpha_i\}$ having the following properties:*

- *A is fully described by $\mathcal{A}$.*
- *$\mathcal{A}$ defines all eigenvalues of A to high relative accuracy; i.e., small relative perturbations $\alpha_i \mapsto \alpha_i(1 + \kappa_i\epsilon)$ cause only small relative perturbations of the eigenvalues.*

EXAMPLE 4.2. *The entries of a bidiagonal matrix B, i.e., $\mathcal{A} = \{a_1, \dots a_n, b_1, \dots, b_{n-1}\}$, form an RRR for all singular values* [4]. *This fact is exploited in the LAPACK implementation of the QR method as well as in the* ***qd*** *and the bisection algorithm* [13, 14, 20].

If the matrix T defining the ***tSEP*** *is one of B^TB, BB^T, or the Golub–Kahan matrix with the entries of B given, then this representation defines all eigenvalues to high relative accuracy.*

By contrast, if the matrix T is given explicitly by its main and first off-diagonals, then these entries generally do not *form an RRR* [9, 10]. *On the other hand, a bi- or tridiagonal structure is not mandatory for having an RRR* [3, 5, 6].

REMARK 4.3 (partial RRRs for indefinite matrices). *Sometimes it is sufficient that $\mathcal{A}$ defines at least a subset of its eigenvalues, e.g., $\lambda_f \geq \cdots \geq \lambda_l$, to high relative accuracy. The RRR algorithm makes heavy use of such* partial RRR*s. There are a priori and a posteriori criteria to find out if a given decomposition of a symmetric tridiagonal matrix forms a (partial) RRR* [9, 10].

In this article we generally assume *that the LDL^T decompositions of certain indefinite matrices form a partial RRR. This is not a severe restriction because decompositions that do not form a suitable partial RRR are rejected later on in the RRR algorithm anyway.*

We do not try to estimate bounds for the eigenvalues' errors. The important point is that under a relative perturbation

$$D, L \xrightarrow{\text{perturbation}} \widehat{D}, \widehat{L},$$

the error remains small, i.e., $\lambda_j = \widehat{\lambda}_j(1 + \kappa\epsilon)$ for $j = f : l$ with a moderate *value for κ.*

4.2. Eigenvalues of separate factorizations. A central building block of the RRR algorithm is the computation of LDL^T (and, more generally, twisted) factorizations of symmetric tridiagonal matrices. What can we conclude from the eigenvalues of the perturbed decompositions of $B^TB - \bar{\mu}^2$ and $BB^T - \bar{\mu}^2$ if they are computed independently from each other?

We first study the situation in exact arithmetic: Suppose that B has a cluster $\sigma_f \geq \cdots \geq \sigma_l$ of singular values with small relative distances and that we choose a shift "close" to one of them:

$$\bar{\mu}^2 = (1 + \kappa_0\epsilon)\sigma_j^2 \quad \text{for some} \quad f \leq j \leq l.$$

Thus the shift parameter $\bar{\mu}$ is a floating point number approximating σ_j. Now we generate $\hat{D}$, $\hat{L}$, $\check{D}$, and $\check{L}$ with Algorithm 1. Thus $\hat{\lambda}_j = \sigma_j^2 - \bar{\mu}^2$ is an eigenvalue of $\hat{L}\hat{D}\hat{L}^T$, and $\check{\lambda}_j = \sigma_j^2 - \bar{\mu}^2$ is an eigenvalue of $\check{L}\check{D}\check{L}^T$. Note that $\hat{\lambda}_j = \check{\lambda}_j = \mathcal{O}(\epsilon\bar{\mu}^2)$.

In floating point arithmetic $\hat{\lambda}_j$ and $\check{\lambda}_j$ typically do not agree. To assess their difference we first apply the mixed stability analysis from Lemma 3.1 to the factorization $B^TB - \bar{\mu}^2 = \hat{L}\hat{D}\hat{L}^T$ computed with Algorithm 1. For the second explicit factorization $BB^T - \bar{\mu}^2 I = \check{L}\check{D}\check{L}^T$ we proceed analogously to Lemma 3.1, introducing perturbations $\check{Z}$ for $\check{Z}$, as well as $\check{D}$ and $\check{L}$ for $\check{D}$ and $\check{L}$. That is, we consider the commuting diagrams

$$\begin{array}{ccccc}
B, (\sigma_j) & \xrightarrow[\text{perturbation}]{\text{initial}} & \hat{Z}, (\hat{\sigma}_j^2) & \xrightarrow[\text{Alg. 1, left}]{\text{computed}} & \hat{D}, \hat{L}, (\hat{\lambda}_j) \\
 & & \Big\downarrow \text{perturbation} & & \Big\uparrow \text{perturbation} \\
 & & \widehat{Z}, (\widehat{\sigma}_j^2) & \xrightarrow[\text{Alg. 1, left}]{\text{exact}} & \widehat{D}, \widehat{L}, (\widehat{\lambda}_j)
\end{array}$$

for the factorization $B^TB - \bar{\mu}^2 I = \hat{L}\hat{D}\hat{L}^T$ and

$$\begin{array}{ccccc} B,\,(\sigma_j) & \xrightarrow[\text{perturbation}]{\text{initial}} & \check{Z},\,(\check{\sigma}_j^2) & \xrightarrow[\text{Alg. 1, right}]{\text{computed}} & \check{D},\check{L},\,(\check{\lambda}_j) \\ & & \Big\downarrow \text{perturbation} & & \Big\uparrow \text{perturbation} \\ & & \breve{Z},\,(\breve{\sigma}_j^2) & \xrightarrow[\text{Alg. 1, right}]{\text{exact}} & \breve{D},\breve{L},\,(\breve{\lambda}_j) \end{array}$$

for the factorization $BB^T - \bar{\mu}^2 I = \check{L}\check{D}\check{L}^T$.

Combining the two mixed analyses we can bound $|\hat{\lambda}_j - \check{\lambda}_j|$; see the following theorem. To be technically precise, we hint at the fact that using $\hat{Z}$ and $\check{Z}$ as input vectors imposes only minimal changes on the corresponding singular values, i.e.,

$$B \xrightarrow[\text{perturbation}]{\text{inital}} \hat{Z} \qquad \Rightarrow \qquad \frac{|\sigma_j - \hat{\sigma}_j|}{|\sigma_j|} = \mathcal{O}(\epsilon).$$

THEOREM 4.4 (on the eigenvalues of separate factorizations). *Let $\hat{D}$ and $\hat{L}$, as well as $\check{D}$ and $\check{L}$, form a partial RRR for the eigenvalues of interest:*

$$\hat{\lambda}_j = \widehat{\lambda}_j \left(1 + \widehat{\kappa}_1^{(j)}\epsilon\right) \quad \text{and} \quad \check{\lambda}_j = \breve{\lambda}_j \left(1 + \breve{\kappa}_1^{(j)}\epsilon\right) \qquad \text{for } j = f:l.$$

Then we have for $j = f:l$

$$\left|\hat{\lambda}_j - \check{\lambda}_j\right| = \mathcal{O}\left(\sigma_j^2\epsilon\right) = \mathcal{O}\left(\bar{\mu}^2\epsilon\right). \tag{4.1}$$

Proof. As bidiagonals always form RRRs for all singular values, the exact singular values of the perturbations $\widehat{Z}$ and $\breve{Z}$ of B (including the initial perturbations) are described by

$$\widehat{\sigma}_j^2 = \sigma_j^2 \left(1 + \widehat{\kappa}_2^{(j)}\epsilon\right) \quad \text{and} \quad \breve{\sigma}_j^2 = \sigma_j^2(1 + \breve{\kappa}_2^{(j)}\epsilon)$$

with suitable $\widehat{\kappa}_2^{(j)}, \breve{\kappa}_2^{(j)} = \mathcal{O}(1)$. By our RRR assumption the eigenvalues of the computed factorizations can similarly be written as

$$\begin{aligned} \hat{\lambda}_j &= \widehat{\lambda}_j \left(1 + \widehat{\kappa}_1^{(j)}\epsilon\right) \\ &= \left(\widehat{\sigma}_j^2 - \bar{\mu}^2\right) \left(1 + \widehat{\kappa}_1^{(j)}\epsilon\right) \\ &= \left(\sigma_j^2 - \bar{\mu}^2\right) \left(1 + \widehat{\kappa}_1^{(j)}\epsilon\right) + \sigma_j^2 \widehat{\kappa}_2^{(j)}\epsilon + \mathcal{O}(\sigma_j^2\epsilon^2) \end{aligned}$$

and

$$\check{\lambda}_j = \left(\sigma_j^2 - \bar{\mu}^2\right) \left(1 + \breve{\kappa}_1^{(j)}\epsilon\right) + \sigma_j^2 \breve{\kappa}_2^{(j)}\epsilon + \mathcal{O}(\sigma_j^2\epsilon^2),$$

respectively. Now (4.1) follows easily, given that $\bar{\mu}^2 \approx \sigma_j^2$. ∎

Note that since $\bar{\mu}$ is chosen close to the cluster of singular values, we have a strong dominance of σ_j^2 compared to $\sigma_j^2 - \bar{\mu}^2 \approx \hat{\lambda}_j \approx \check{\lambda}_j$. Therefore $\sigma_j^2 - \bar{\mu}^2$ and $\hat{\lambda}_j$ may differ

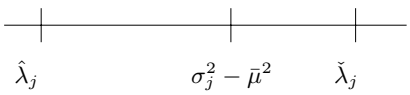


FIG. 4.1. *The eigenvalues of separate factorizations typically have a large absolute deviation* $|\hat{\lambda}_j - \check{\lambda}_j| = \mathcal{O}(\bar{\mu}^2\epsilon)$.

TABLE 4.1

Comparison of the eigenvalue approximations. For $j = 1, 3, \ldots, 13$ *the third and fourth columns list the approximations to the* j*th and* $(j+1)$*st eigenvalues of* $\hat{L}\hat{D}\hat{L}^T$ *and* $\check{L}\check{D}\check{L}^T$*. The numbers of the second column which are marked with a* $*$ *are used as shift parameter* $\bar{\mu}^2$ *for the respective factorizations.*

j	σ_j^2	$\hat{\lambda}_j$	$\check{\lambda}_j$
1	$*$12.748194182904	-1.8608310027439D-14	-1.8992680959841D-14
2	12.748194182904	-9.0249074144178D-14	-9.0667129038281D-14
3	$*$11.212678647362	-1.2794598835728D-14	-1.2744594323512D-14
4	11.212678647305	-5.6426308574546D-11	-5.6426258580880D-11
5	$*$10.040941122829	-1.2922467590762D-14	-1.2999176683819D-14
6	10.040941115815	-7.0147628503429D-09	-7.0147629270521D-09
7	$*$9.005952209529	-1.4882896339264D-14	-1.4366985384200D-14
8	9.005951798617	-4.1091231558744D-07	-4.1091231507153D-07
9	$*$8.002234031585	-1.2863778785396D-14	-1.2488543209404D-14
10	8.002217522257	-1.6509327081740D-05	-1.6509327081365D-05
11	7.002244425002	4.6194725901782D-04	4.6194725901810D-04
12	$*$7.001782477743	6.6765339180722D-15	6.9574116568442D-15
13	6.006354023441	8.3058220572403D-03	8.3058220572403D-03
14	$*$5.998048201384	8.6013525452083D-15	8.6448127042717D-15

in all significant digits. Thus $\sigma_j^2 - \bar{\mu}^2$ typically is not a good choice to approximate the eigenvalues of $\hat{L}\hat{D}\hat{L}^T$ to high relative accuracy. Figure 4.1 illustrates the situation.

Theorem 4.4 says that the smallest eigenvalues of $\hat{L}\hat{D}\hat{L}^T$ and $\check{L}\check{D}\check{L}^T$ may have large *absolute* deviations. We point out that the differences are tied to the exact eigenvalues of the perturbed decompositions. Thus they are solely induced by the rounding errors that occur if we compute the factorizations separately with Algorithm 1—numerical computation of the eigenvalues is not involved in any way. We illustrate the negative impact of absolute deviations with an example.

EXAMPLE 4.5 (the Wilkinson matrix Wilk_{21}^+). *A standard test problem for the computation of eigensystems is given by*

$$\begin{aligned}\text{Wilk}_{21}^+ = {} & \texttt{diag}([10, 9, 8, \ldots, 8, 9, 10]) \\ & + \texttt{diag}([1, \ldots, 1], 1) + \texttt{diag}([1, \ldots, 1], -1).\end{aligned}$$

Although it is of moderate condition and unreduced it has pairs of eigenvalues which lie very close, i.e., agreeing to a considerable number of digits [23]*. Using a shift parameter* $\tau < \lambda_{21}$*, we can get an upper bidiagonal matrix* B *as Cholesky factor regarding* $\text{Wilk}_{21}^+ - \tau I = B^T B$*. The singular values of* B *can be computed to high relative accuracy (e.g., using the QR method or the* ***qd*** *algorithm). The approximations to their squares—the eigenvalues of* $B^T B$ *and* BB^T*—are shown in the second column of Table* 4.1*.*

The spectrum is structured in a number of close pairs of eigenvalues. We choose one eigenvalue approximation from each cluster for the shift $\bar{\mu}^2$ *and form*

$$B^T B - \bar{\mu}^2 I = \hat{L}\hat{D}\hat{L}^T \qquad \text{and} \qquad BB^T - \bar{\mu}^2 I = \check{L}\check{D}\check{L}^T$$

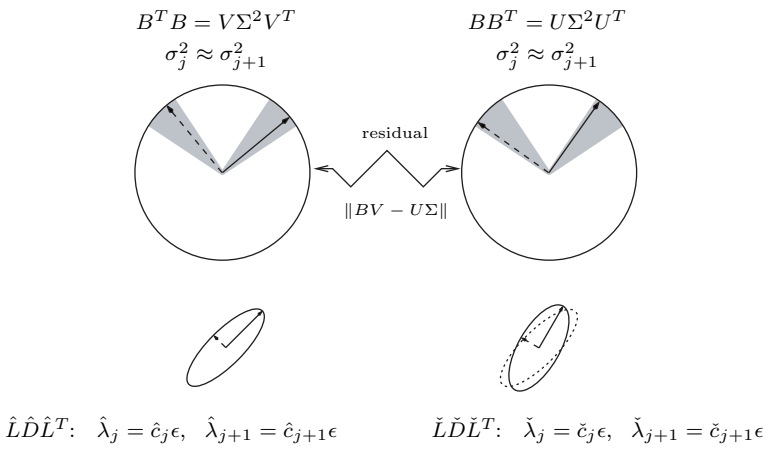


FIG. 4.2. *Round and stretched ellipsoids. The grey segments symbolize the area of uncertainty. The stretched ellipsoids are well suited to compute orthogonal basis vectors, but the large absolute deviations of the new eigenvalues lead to poor couplings. Thus the residual* $\|B\bar{V} - \bar{U}\bar{\Sigma}\|$ *for the computed approximations is typically too large. Note that due to* $|\hat{\lambda}_j| = \mathcal{O}(\epsilon\sigma_j^2)$ *the stretched ellipsoids are by several orders of magnitude smaller than the round ones; in our pictures they are heavily magnified.*

separately with the left- and right-hand sides of Algorithm 1. *It is possible to verify numerically (e.g., by checking relative condition numbers a posteriori* [9, 10]*; see also Remark* 4.3*) that all factorizations form a partial RRR. Thus the computed eigenvalues in the third and fourth columns of Table* 4.1 *represent the exact eigenvalues of the factorizations with high relative accuracy. Comparing these approximations to* $\hat{\lambda}_j$ *and* $\check{\lambda}_j$ *manifests large deviations particularly for small eigenvalues. This is exactly the behavior predicted by the absolute deviation bound given in* (4.1).

4.3. A geometric interpretation. According to the Courant–Fischer minimax theorem [17], the computation of an eigenvector q_j corresponding to λ_j can be interpreted as finding an extremal point on a projected ellipsoid given by the symmetric matrix A:

$$\lambda_j = \max_{\substack{\|x\|_2=1 \\ x\perp \operatorname{span}\{q_1,\dots,q_{j-1}\}}} x^TAx \qquad \text{for } x = q_j.$$

The projection of the ellipsoid onto the subspace corresponding to a tight cluster of eigenvalues is almost a perfect sphere. Finding the solutions for the extremal points of such "round" ellipsoids is the main challenge for all methods computing eigenvector approximations, because in floating point arithmetic we typically cannot reach the extremal points exactly. The best we can achieve is to construct a numerically orthogonal basis, but this basis is by no means unique even if the eigenvalues are distinct. There is a significant amount of freedom for choosing such a basis; see the shaded segments in Figure 4.2.

Suppose that the RRR algorithm is applied to compute eigenvector approximations for $\mathrm{Wilk}_{21}^+ - \tau I = B^TB$ and BB^T separately using the shift parameters marked with the $*$ from Table 4.1 for each cluster. A key idea of the method can be formulated as follows: *Eigenvectors are shift-invariant, whereas relative distances are not.* Using the shifted matrices $B^TB - \bar{\mu}^2I = \hat{L}\hat{D}\hat{L}^T$ and $BB^T - \bar{\mu}^2I = \check{L}\check{D}\check{L}^T$ means working

on "stretched" ellipsoids because, noting that

$$\frac{|\hat{\lambda}_j - \hat{\lambda}_{j+1}|}{|\hat{\lambda}_j|} \gg \frac{|\sigma_j^2 - \sigma_{j+1}^2|}{|\sigma_j^2|},$$

we typically have sufficiently large relative distances between the new eigenvalues $\hat{\lambda}_j$ and $\hat{\lambda}_{j+1}$, as well as between $\check{\lambda}_j$ and $\check{\lambda}_{j+1}$. Thus we can easily compute numerically orthogonal bases of the subspaces for the left and for the right singular vectors from the stretched ellipsoids. In this way we have found a good basis approximation for the original round ellipsoids, too.

Although this strategy is fully adequate for the respective **tSEP**s, we point out that the choice of these orthogonal bases strongly depends on the new eigenvalues of the computed matrices $\hat{L}\hat{D}\hat{L}^T$ and $\check{L}\check{D}\check{L}^T$. Due to the large absolute deviations of these eigenvalues (4.1) it is obvious that these bases can be poorly coupled: The stretched ellipsoids for the right and left subspaces—defined by the computed matrices and $\{\hat{\lambda}_j, \hat{\lambda}_{j+1}\}$ and $\{\check{\lambda}_j, \check{\lambda}_{j+1}\}$—may differ significantly. This situation is illustrated in the lower right picture of Figure 4.2, where the stretched ellipsoids for $\hat{L}\hat{D}\hat{L}^T$ (dotted line) and $\check{L}\check{D}\check{L}^T$ (solid line) are sketched.

Some of the approximated singular vector pairs from Example 4.5 do in fact show such an insufficient quality with respect to the residual, e.g, in double precision ($\epsilon \approx 2 \cdot 10^{-16}$) we observe $\bar{U}(:,16)^T B \bar{V}(:,17) = \mathcal{O}(10^{-7})$ and even $\bar{U}(:,20)^T B \bar{V}(:,21) = \mathcal{O}(10^{-2})$, whereas both values should be zero.

5. The impact of couplings. This section describes how to construct representations of translates of $B^T B$ and BB^T which preserve small residuals of the approximated singular vectors. The key idea is to form an explicit LDL^T factorization of translates of the Golub–Kahan matrix T_{GK} and to use coupling transformations to set up the corresponding decompositions for the normal equations (cf. the horizontal arrows in Figure 2.1). Theorem 5.2 shows that the eigenvalues of these coupled factorizations agree to high *relative* accuracy. Moreover, in the case of positive definite initial matrices it is possible to convert directly from $[\hat{D}, \hat{L}]$ to $[\check{D}, \check{L}]$ in a stable way without accessing the data from T_{GK}.

5.1. Couplings of the diagonal pivot elements. In exact arithmetic the LDL^T decompositions of (translates of) $B^T B$, BB^T, and T_{GK} can be related by a set of coupling transformations stated in the following lemma.

LEMMA 5.1 (coupling $[\hat{D}^+, \hat{L}^+] \leftrightarrow [\tilde{D}^+, \tilde{L}^+] \leftrightarrow [\check{D}^+, \check{L}^+]$). *If the factorizations* (2.2)–(2.4) *exist, we have for* $i = 1 : n-1$*:*

$$\begin{aligned} \hat{d}_i^+ &= -\tilde{d}_{2i-1}^+ \tilde{d}_{2i}^+, & \hat{d}_n^+ &= -\tilde{d}_{2n-1}^+ \tilde{d}_{2n}^+, & \hat{l}_i^+ &= -\tilde{l}_{2i-1}^+ \tilde{l}_{2i}^+, \\ \check{d}_i^+ &= -\tilde{d}_{2i}^+ \tilde{d}_{2i+1}^+, & \check{d}_n^+ &= -\tilde{d}_{2n}^+ \tilde{d}_1^+, & \check{l}_i^+ &= -\tilde{l}_{2i}^+ \tilde{l}_{2i+1}^+. \end{aligned}$$

See [19] for a proof.

Thus we can relate the diagonal pivots, d_i (as well as the off-diagonal elements of the bidiagonal factors, l_i), using only multiplications. If we can compute an RRR for a translate of the Golub–Kahan matrix, it is straightforward to set up a backward stable algorithm to find representations for the normal equations; cf. Figure 5.1. Note that the relations between the diagonal pivots are also valid for the factorizations in (2.1).

To characterize the quality of the coupled eigenvalues we briefly recall the situation in exact arithmetic. For the jth eigenvalue of $\tilde{L}^+ \tilde{D}^+ (\tilde{L}^+)^T$ we have $\tilde{\lambda}_j^+ = \sigma_j -$

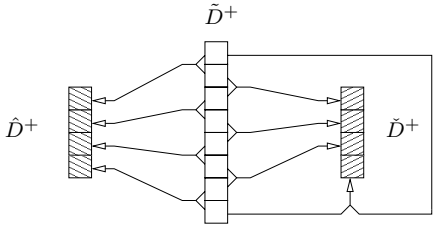


FIG. 5.1. *Data flow for the conversion* $[\tilde{D}^+] \to [\hat{D}^+, \check{D}^+]$.

$(\bar{\mu}+\mu)$. The corresponding eigenvalues $\hat{\lambda}_j^+$ and $\check{\lambda}_j^+$ of $\hat{L}^+\hat{D}^+(\hat{L}^+)^T$ and $\check{L}^+\check{D}^+(\check{L}^+)^T$, respectively, can be written as

$$\hat{\lambda}_j^+ = \check{\lambda}_j^+ = \sigma_j^2 - (\bar{\mu}+\mu)^2 = \tilde{\lambda}_j^+(\sigma_j + (\bar{\mu}+\mu)) = \tilde{\lambda}_j^+(2(\bar{\mu}+\mu) + \tilde{\lambda}_j^+).$$

The effects of floating point arithmetic are discussed in the following theorem.

THEOREM 5.2 (on the eigenvalues of coupled factorizations). *Let* $\tilde{L}^+\tilde{D}^+(\tilde{L}^+)^T$ *form a partial RRR for the eigenvalues of interest and suppose that* $\hat{D}^+$, $\hat{L}^+$, $\check{D}^+$, *and* $\check{L}^+$ *are computed according to Lemma* 5.1. *Then both* $\hat{L}^+\hat{D}^+(\hat{L}^+)^T$ *and* $\check{L}^+\check{D}^+(\check{L}^+)^T$ *also form a partial RRR, and we can relate the eigenvalues via*

$$\hat{\lambda}_j^+ = \tilde{\lambda}_j^+ \left(2(\bar{\mu}+\mu) + \tilde{\lambda}_j^+\right)(1+\hat{\kappa}\epsilon) \tag{5.1}$$

and

$$\check{\lambda}_j^+ = \tilde{\lambda}_j^+ \left(2(\bar{\mu}+\mu) + \tilde{\lambda}_j^+\right)(1+\check{\kappa}\epsilon), \tag{5.2}$$

where $\hat{\kappa}, \check{\kappa} = \mathcal{O}(1)$. *Thus we have a small relative deviation*

$$\left|\frac{\hat{\lambda}_j^+ - \check{\lambda}_j^+}{\hat{\lambda}_j^+}\right| = \epsilon \left|\frac{\hat{\kappa}-\check{\kappa}}{1+\hat{\kappa}\epsilon}\right| = \mathcal{O}(\epsilon). \tag{5.3}$$

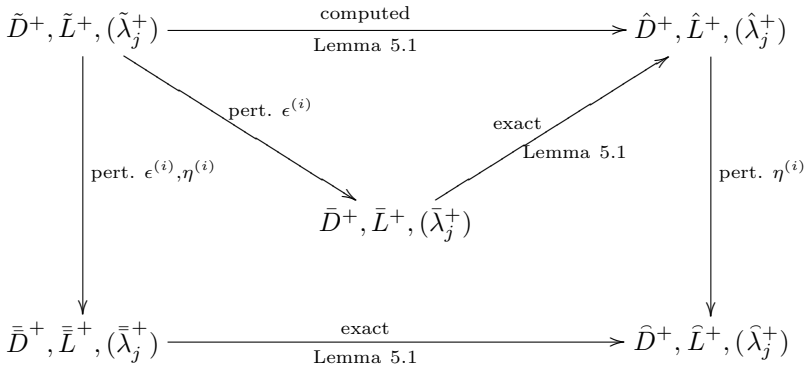
Proof. Performing the computations of Lemma 5.1 in floating point arithmetic yields

$$\begin{aligned} \hat{d}_i^+ &= -\tilde{d}_{2i-1}^+\tilde{d}_{2i}^+\left(1+\epsilon_1^{(i)}\right), \qquad i = 1:n, \\ \hat{l}_i^+ &= -\tilde{l}_{2i-1}^+\tilde{l}_{2i}^+\left(1+\epsilon_2^{(i)}\right), \qquad i = 1:n-1 \end{aligned}$$

with $|\epsilon_1^{(i)}|, |\epsilon_2^{(i)}| \le \epsilon$. Therefore the computed data $\hat{D}^+$, $\hat{L}^+$ could also have been obtained by applying Lemma 5.1 in *exact* arithmetic to a slightly perturbed decomposition $\bar{L}^+\bar{D}^+(\bar{L}^+)^T$, where

$$\begin{aligned} \bar{d}_{2i-1}^+ &= \tilde{d}_{2i-1}^+, \quad \bar{d}_{2i}^+ = \tilde{d}_{2i}^+\left(1+\epsilon_1^{(i)}\right), \qquad i = 1:n, \\ \bar{l}_{2i-1}^+ &= \tilde{l}_{2i-1}^+, \quad \bar{l}_{2i}^+ = \tilde{l}_{2i}^+\left(1+\epsilon_2^{(i)}\right), \qquad i = 1:n-1, \end{aligned}$$

and $\bar{l}_{2n-1}^+ = \tilde{l}_{2n-1}^+$; see the following diagram.



By the RRR property of $\tilde{L}^+\tilde{D}^+(\tilde{L}^+)^T$ this perturbation only slightly changes the eigenvalues of interest:

$$\bar{\lambda}_j^+ = \tilde{\lambda}_j^+(1+\kappa_1\epsilon) \quad \text{with } \kappa_1 = \mathcal{O}(1).$$

As the transition $[\bar{D}^+, \bar{L}^+] \longrightarrow [\hat{D}^+, \hat{L}^+]$ is exact, we have

$$\begin{aligned}\hat{\lambda}_j^+ &= \bar{\lambda}_j^+ \left(2(\bar{\mu}+\mu)+\bar{\lambda}_j\right)\\ &= \tilde{\lambda}_j^+(1+\kappa_1\epsilon)\left(2(\bar{\mu}+\mu)+\tilde{\lambda}_j^+(1+\kappa_1\epsilon)\right)\\ &= \tilde{\lambda}_j^+\left(2(\bar{\mu}+\mu)+\tilde{\lambda}_j^+\right)\left[1+\kappa_1\epsilon+\frac{\tilde{\lambda}_j^+(1+\kappa_1\epsilon)\kappa_1}{2(\bar{\mu}+\mu)+\tilde{\lambda}_j^+}\epsilon\right].\end{aligned}$$

Now, since $\bar{\mu}+\mu$ was chosen as an approximation to σ_j, we have

$$|\tilde{\lambda}_j^+| \approx |\sigma_j - (\bar{\mu}+\mu)| \ll |\sigma_j + (\bar{\mu}+\mu)| \approx |\tilde{\lambda}_j^+ + 2(\bar{\mu}+\mu)|,$$

and therefore setting

$$\hat{\kappa} := \kappa_1 + \frac{\tilde{\lambda}_j^+(1+\kappa_1\epsilon)\kappa_1}{2(\bar{\mu}+\mu)+\tilde{\lambda}_j^+}$$

immediately shows (5.1).

Note that a similar argument also holds for an arbitrary perturbation

$$\begin{aligned}\hat{d}_i^+ &\mapsto \hat{d}_i^+\left(1+\eta_1^{(i)}\right) =: \widehat{d}_i^+, \qquad i = 1:n,\\ \hat{l}_i^+ &\mapsto \hat{l}_i^+\left(1+\eta_2^{(i)}\right) =: \widehat{l}_i^+, \qquad i = 1:n-1.\end{aligned}$$

By introducing

$$\begin{aligned}\bar{\bar{d}}_{2i-1}^+ &= \tilde{d}_{2i-1}^+, \qquad \bar{\bar{d}}_{2i}^+ = \tilde{d}_{2i}^+\left(1+\epsilon_1^{(i)}\right)\left(1+\eta_1^{(i)}\right), \qquad i = 1:n,\\ \bar{\bar{l}}_{2i-1}^+ &= \tilde{l}_{2i-1}^+, \qquad \bar{\bar{l}}_{2i}^+ = \tilde{l}_{2i}^+\left(1+\epsilon_2^{(i)}\right)\left(1+\eta_2^{(i)}\right), \qquad i = 1:n-1,\end{aligned}$$

and $\bar{\bar{l}}_{2n-1}^+ = \tilde{l}_{2n-1}^+$, we see that the eigenvalues of $\widehat{L}^+\widehat{D}^+(\widehat{L}^+)^T$ can also be written in the form

$$\widehat{\lambda}_j^+ = \tilde{\lambda}_j^+\left(2(\bar{\mu}+\mu)+\tilde{\lambda}_j^+\right)(1+\hat{\kappa}\epsilon).$$

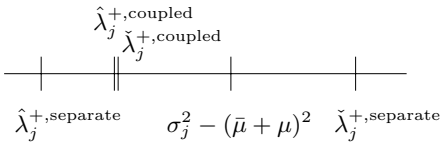


FIG. 5.2. *The eigenvalues of the coupled representations agree to high relative accuracy in contrast to those produced by separate factorizations.*

Together with (5.1) this implies that $\widehat{\lambda}_j^+$ and $\hat{\lambda}_j^+$ agree up to a small relative difference, and thus $\hat{L}^+\hat{D}^+(\hat{L}^+)^T$ is shown to be an RRR.

The proof for (5.2) and for $\check{L}^+\check{D}^+(\check{L}^+)^T$ being an RRR is completely analogous, and (5.3) follows immediately from (5.1) and (5.2). ∎

The key message of Theorem 5.2 is that the exact eigenvalues $\hat{\lambda}_j^+$ and $\check{\lambda}_j^+$ of the matrices $\hat{L}^+\hat{D}^+(\hat{L}^+)^T$ and $\check{L}^+\check{D}^+(\check{L}^+)^T$ generated by the coupling transformation of Lemma 5.1 do agree to almost all digits; see Figure 5.2.

We now revisit the geometric interpretation of section 4.3 with the new approach: First form an explicit factorization of the Golub–Kahan matrix and then determine representations for the normal equations using Lemma 5.1. Theorem 5.2 says that the new eigenvalues may have at most a small relative deviation. Thus we can conclude that the stretched projected ellipsoids are nearly identical, and therefore the choice of the bases leads to well-coupled singular vector pairs. Numerical orthogonality is still intact: If a new eigenvalue $\tilde{\lambda}_j^+$ derived from T_{GK} is isolated, then this also holds for the new eigenvalues of the coupled representations. Note that this cannot be guaranteed in the case of separate factorizations because we may have different shift parameters at deeper recursion levels when applying the RRR algorithm to B^TB and BB^T as a black box.

We point out that a black box application to the Golub–Kahan matrix itself may lead to problems for matrices B having a large condition number and clusters of tiny singular values: Factorizing T_{GK} with tiny shift parameters typically results in diagonal pivots which strongly alternate from very large to very small magnitudes [18]. Thus an LDL^T decomposition computed with Algorithms 2 and 1 is unlikely to form a partial RRR. An alternative would be to use a block decomposition

$$P_{\text{block}}(T_{\text{GK}} - (\bar{\mu}+\mu)I)P_{\text{block}}^T = MDM^T,$$

where P_{block} is a suitable permutation and D is block diagonal with blocks of order one or two [2].

If we consider the corresponding LDL^T decompositions for the normal equations, the problem of strongly alternating pivot elements vanishes: Although we have large fluctuations for two consecutive elements $\tilde{d}_{2i-1}^+$ and $\tilde{d}_{2i}^+$, their pairwise products—the diagonal pivots derived from the normal equations—typically remain at moderate values.

There is another unfavorable consequence of strongly alternating pivot elements when applying the RRR algorithm to T_{GK} solely. Although it generates an adequate approximation to the eigenvector basis $Q(:, n+1:2n) \in \mathbb{R}^{2n\times n}$, the approximations to U and V extracted from the even and odd rows turn out to have sufficient quality regarding the residual, but they may be far from numerical orthogonality.

5.2. Couplings for positive definite initial matrices. As seen in the previous section there are cases where it is preferable to work on the normal equations and

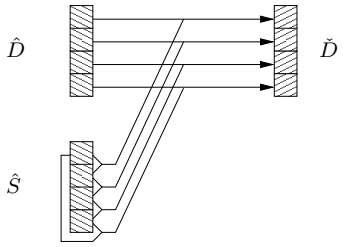


FIG. 5.3. *Data flow for the conversion* $\hat{D} \to \check{D}$.

to abstain from decompositions based on the Golub–Kahan matrix. To this aim we formulate coupling transformations for positive definite initial matrices.

LEMMA 5.3 (coupling $\hat{D} \leftrightarrow \check{D}$ directly). *Supposing that the factorizations in* (2.1) *can be computed using Algorithms* 1 *and* 2*, we have for* $i = 1 : n$

$$\bar{\mu}\tilde{d}_{2i-1} = \hat{s}_i \quad \text{and} \quad \bar{\mu}\tilde{d}_{2i} = \check{p}_i. \tag{5.4}$$

Moreover, setting $\hat{s}_{n+1} = \hat{s}_1 = -\bar{\mu}^2$ *and* $\check{d}_0 = \check{p}_0 = 1$*, we have for* $i = 1 : n$

$$\check{d}_i = \hat{s}_{i+1}\frac{\hat{d}_i}{\hat{s}_i} \quad \text{and} \quad \hat{d}_i = \check{p}_i\frac{\check{d}_{i-1}}{\check{p}_{i-1}}. \tag{5.5}$$

See [13] for a proof of (5.4) and [19] for (5.5).

Using Lemma 5.3 we can set up a transformation $\hat{D} \to \check{D}$ which connects the data from the LDL^T decompositions of $B^TB - \bar{\mu}^2 I$ and $BB^T - \bar{\mu}^2 I$ directly using only multiplications and divisions; cf. Figure 5.3.

Theorem 4.4 gave an *absolute* bound for the difference of eigenvalues of separate factorizations:

$$\left|\hat{\lambda}_j - \check{\lambda}_j\right| = \mathcal{O}(\bar{\mu}^2\epsilon) \quad \text{for} \quad j = f : l.$$

Again, couplings provide a major improvement. Suppose that $[\hat{D}, \hat{L}]$ is formed by an explicit factorization with the left-hand side of Algorithm 1, whereas $[\check{D}, \check{L}]$ is constructed using Lemma 5.3. The next theorem shows that then

$$\left|\frac{\hat{\lambda}_j - \check{\lambda}_j}{\hat{\lambda}_j}\right| = \mathcal{O}(\epsilon) \quad \text{for} \quad j = f : l.$$

Thus the eigenvalues of $\hat{L}\hat{D}\hat{L}^T$ and $\check{L}\check{D}\check{L}^T$ have a small *relative* distance.

Note that as in Theorem 4.4, using $\hat{Z}$ as input vector imposes a small initial perturbation on the singular values.

THEOREM 5.4 (on the eigenvalues of coupled factorizations). *We assume that* $[\hat{D}, \hat{L}]$ *forms a partial RRR for the eigenvalues of interest and that* $[\check{D}, \check{L}]$ *is computed according to Lemma* 5.3*. Then* $[\check{D}, \check{L}]$ *also forms a partial RRR and we have*

$$\check{\lambda}_j = \hat{\lambda}_j(1 + \check{\kappa}^{(j)}\epsilon) \quad \text{for } j = f : l.$$

Proof. As the computation $[\hat{D}, \hat{L}] \longrightarrow [\check{D}, \check{L}]$ involves only multiplications and divisions, the backward error analysis we gave in the proof of Theorem 5.2 carries over—with straightforward modifications—to this case. ∎

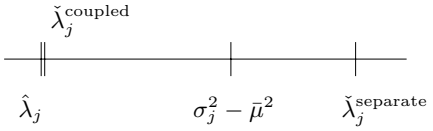


FIG. 5.4. *Location of the eigenvalues using the direct couplings from Lemma* 5.3.

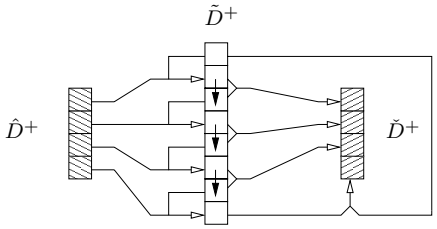


FIG. 5.5. *Data flow for the conversion* $[\hat{D}^+] \to [\tilde{D}^+_{\text{even}}] \to [\tilde{D}^+_{\text{odd}}] \to [\check{D}^+]$.

Using the auxiliary variables from the differential **qd** transformations allows us to couple the LDL^T decompositions for positive definite initial matrices directly. The eigenvalues of these coupled representations do agree to high relative accuracy; cf. Figure 5.4 and Table 2.1. Together with the geometric interpretation from section 4.3 we can conclude that the corresponding singular vector pairs are well coupled.

5.3. Coupling $[\hat{\boldsymbol{D}}^+] \to [\tilde{\boldsymbol{D}}^+_{\textbf{even}}] \to [\tilde{\boldsymbol{D}}^+_{\textbf{odd}}] \to [\check{\boldsymbol{D}}^+]$. We finally discuss how to transform $\hat{D}^+ \to \check{D}^+$ for indefinite initial matrices.

Therefore we suppose that the factorization $\hat{L}\hat{D}\hat{L}^T - \nu I = \hat{L}^+\hat{D}^+(\hat{L}^+)^T$ from (2.2) is given explicitly. Here, we cannot utilize the auxiliary variables from the differential **qd** transformations generated by Algorithm 3 to find direct couplings $\hat{D}^+ \to \check{D}^+$ which are backward stable. Instead we have to perform a partial factorization of $\tilde{L}\tilde{D}\tilde{L}^T - \mu I$: We can exploit the data of $\hat{D}^+$ to compute the even-numbered elements of $\tilde{D}^+$ (arrows with empty heads in Figure 5.5). But the successive odd-numbered elements have to be determined by applying one step of Algorithm 3 or the classical factorization procedure (arrows with filled heads in Figure 5.5) [19].

Theorem 5.2 assures that if $\tilde{L}^+\tilde{D}^+(\tilde{L}^+)^T$ forms a partial RRR, then both $\hat{L}^+\hat{D}^+(\hat{L}^+)^T$ and $\check{L}^+\check{D}^+(\check{L}^+)^T$ also form a partial RRR. On the other hand, it is not clear if the property of $\hat{L}^+\hat{D}^+(\hat{L}^+)^T$ to form a partial RRR implies that $\tilde{L}^+\tilde{D}^+(\tilde{L}^+)^T$ also shares this feature. (We have mentioned the large changes in the diagonal pivots when factorizing a translate of the Golub–Kahan matrix in the case of tiny and clustered singular values in section 5.1.) Since we cannot find a backward stable transformation from $\hat{L}^+\hat{D}^+(\hat{L}^+)^T$ to $\check{L}^+\check{D}^+(\check{L}^+)^T$ in the case of indefinite initial matrices, it is hard to give theoretical criteria to decide if the eigenvalues of the latter representation can be determined to high relative accuracy and additionally lie close to those of $\hat{L}^+\hat{D}^+(\hat{L}^+)^T$. Thus we can control the deviation of the eigenvalues only a posteriori, e.g., by comparing the eigenvalues determined with a highly accurate numerical approximation procedure.

5.4. Using the couplings in the RRR algorithm. We finally propose a strategy to find LDL^T decompositions which form partial relatively robust representations and in addition are *well coupled*.

5.4.1. Positive definite initial matrices. Here, the only requirement is to find a shift parameter $\bar{\mu}$ such that the explicit factorization $B^TB-\bar{\mu}^2I = \hat{L}\hat{D}\hat{L}^T$ computed with Algorithm 1 forms a partial RRR for the eigenvalues of interest. We then use Lemmas 5.3 and 5.1 to compute the data of $\tilde{L}\tilde{D}\tilde{L}^T$ and $\check{L}\check{D}\check{L}^T$ implicitly. Theorem 5.4 says that the latter matrices form a partial RRR and that the respective eigenvalues lie close. In a computer implementation we then can determine approximations to $\hat{\lambda}_f, \ldots, \hat{\lambda}_l$, which in turn also yield precise guesses for the eigenvalues of the two coupled factorizations.

5.4.2. Indefinite initial matrices. If we can show that the decomposition $\tilde{L}\tilde{D}\tilde{L}^T - \mu I = \tilde{L}^+\tilde{D}^+(\tilde{L}^+)^T$ based on the Golub–Kahan matrix forms a partial RRR, e.g., by checking the a priori criteria proposed in [9, 10], we can proceed as follows: We use Lemma 5.1 to determine the data $[\hat{D}^+, \hat{L}^+]$ and $[\check{D}^+, \check{L}^+]$. According to Theorem 5.2, these quantities form a partial RRR. With respect to performance issues (length of inner loops), a computer implementation should avoid computing $\tilde{\lambda}_f^+, \ldots, \tilde{\lambda}_l^+$ and approximate $\hat{\lambda}_f^+, \ldots, \hat{\lambda}_l^+$ instead.

In some cases it is easier to show that the factorizations of the normal equations form a partial RRR, while the quality of $[\tilde{D}^+, \tilde{L}^+]$ is unknown. We then compute $\hat{L}\hat{D}\hat{L}^T - \nu I = \hat{L}^+\hat{D}^+(\hat{L}^+)^T$ explicitly and use the scheme described in Figure 5.5 to determine the coupled data $[\tilde{D}^+, \tilde{L}^+]$ and $[\check{D}^+, \check{L}^+]$. We then compute highly accurate approximations to $\hat{\lambda}_f^+, \ldots, \hat{\lambda}_l^+$ as well as to $\check{\lambda}_f^+, \ldots, \check{\lambda}_l^+$. The deviation of the respective eigenvalues is thus controlled a posteriori and numerically. Various experiments show that in most cases the eigenvalues are very close and the couplings are adequate even if this could not be proved a priori. Nevertheless, there are few cases where the coupling between $\hat{L}^+\hat{D}^+(\hat{L}^+)^T$ and $\check{L}^+\check{D}^+(\check{L}^+)^T$ has to be considered to be insufficient. Then we can try varying the shift parameter and restart.

5.4.3. Implementation. Embedding the coupling transformations does not require substantial changes of the algorithmic structure of the RRR procedure for symmetric tridiagonal matrices as implemented, e.g., in the LAPACK routine `DSTEGR` [1]. Whenever an LDL^T factorization is computed there, we can easily add a call to a routine performing the coupling transformations.

Note that `DSTEGR` typically can resolve most of the eigenvalue clusters by shifting only once. Thus the initial matrices are positive definite in the majority of cases, and we can use Lemma 5.3 for the couplings.

6. Conclusions. This article provides a theoretical framework explaining how to compute singular vector pairs of a bidiagonal matrix B efficiently. This task is closely connected to the symmetric eigenproblems given by the tridiagonals B^TB, BB^T, and T_{GK}.

If translates of these three matrices are factorized separately according to (2.1), roundoff errors cause large, i.e., *absolute*, deviations of the corresponding new eigenvalues of the shifted representations. Using a geometric interpretation (stretched ellipsoids, area of uncertainty; cf. Figure 4.2) we explain why the singular vector pairs are poorly coupled in this case.

As a solution we relate the factorizations implicitly by a set of coupling transformations. We prove that the eigenvalues of these coupled representations have small, i.e., *relative*, deviations leading to well-coupled singular vector pairs. These results are used for generalizing the RRR algorithm to the solution of the bidiagonal SVD.

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., SIAM, Philadelphia, 1999.

[2] J. Bunch, L. Kaufman, and B. Parlett, *Decomposition of a symmetric matrix*, Numer. Math., 27 (1976), pp. 95–107.

[3] J. Demmel, M. Gu, S. Eisenstat, I. Slapnicar, K. Veselić, and Z. Drmač, *Computing the singular value decomposition with high relative accuracy*, Linear Algebra Appl., 299 (1992), pp. 21–80.

[4] J. Demmel and W. Kahan, *Accurate singular values of bidiagonal matrices*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 873–912.

[5] J. Demmel and P. Koev, *The accurate and efficient solution of a totally positive generalized Vandermonde linear system*, SIAM J. Matrix Anal. Appl., to appear.

[6] J. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.

[7] I. Dhillon and B. Parlett, *Fernando's solution to Wilkinson's problem: An application of double factorization*, Linear Algebra Appl., 267 (1997), pp. 247–279.

[8] I. S. Dhillon and B. N. Parlett, *Orthogonal eigenvectors and relative gaps*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 858–899.

[9] I. Dhillon and B. Parlett, *Relatively robust representations for symmetric tridiagonals*, Linear Algebra Appl., 309 (2000), pp. 121–151.

[10] I. S. Dhillon, *A new $\mathcal{O}(n^2)$ Algorithm for the Symmetric Tridiagonal Eigenvalue/Eigenvector Problem*, Ph.D. thesis, University of California, Berkeley, CA, 1997.

[11] K. Fernando, *Computing an eigenvector of a tridiagonal when the eigenvalue is known*, Z. Angew. Math. Mech., 76 (1996), pp. 299–302.

[12] K. V. Fernando, *On computing an eigenvector of a tridiagonal matrix. Part* I*: Basic results*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 1013–1034.

[13] K. V. Fernando, *Accurately counting singular values of bidiagonal matrices and eigenvalues of skew-symmetric tridiagonal matrices*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 373–399.

[14] K. Fernando and B. Parlett, *Accurate singular values and differential qd algorithms*, Numer. Math., 67 (1994), pp. 191–229.

[15] S. Godunov, A. Antonov, O. Kiriljuk, and V. Kostin, *Guaranteed Accuracy in Numerical Linear Algebra*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1993.

[16] S. Godunov, V. Kostin, and A. Mitchenko, *Computation of an eigenvector of a symmetric tridiagonal matrix*, Siberian Math. J., 26 (1985), pp. 684–696.

[17] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[18] B. Grosser, *Ein paralleler und hochgenauer $\mathcal{O}(n^2)$ Algorithmus für die bidiagonale Singulärwertzerlegung*, Ph.D. thesis, Bergische Universität GH Wuppertal, Fachbereich Mathematik, Wuppertal, Germany, 2001.

[19] B. Grosser and B. Lang, *An $\mathcal{O}(n^2)$ algorithm for the bidiagonal SVD*, Linear Algebra Appl., 358 (2003), pp. 45–70.

[20] O. Marques and B. Parlett, *An implementation of the dqds algorithm (positive case)*, Linear Algebra Appl., 309 (2000), pp. 217–259.

[21] H. Rutishauser, *Der Quotienten-Differenzen-Algorithmus*, Z. Angew. Math. Phys., 5 (1954), pp. 233–251.

[22] C. Van Loan, *Computational Frameworks for the Fast Fourier Transform*, Frontiers Appl. Math. 10, SIAM, Philadelphia, 1992.

[23] J. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 3, pp. 621–639

ROW MODIFICATIONS OF A SPARSE CHOLESKY FACTORIZATION*

TIMOTHY A. DAVIS† AND WILLIAM W. HAGER‡

Abstract. Given a sparse, symmetric positive definite matrix $\mathbf{C}$ and an associated sparse Cholesky factorization $\mathbf{LDL}^\mathsf{T}$, we develop sparse techniques for updating the factorization after a symmetric modification of a row and column of $\mathbf{C}$. We show how the modification in the Cholesky factorization associated with this rank-2 modification of $\mathbf{C}$ can be computed efficiently using a sparse rank-1 technique developed in [T. A. Davis and W. W. Hager, *SIAM J. Matrix Anal. Appl.*, 20 (1999), pp. 606–627]. We also determine how the solution of a linear system $\mathbf{L}\boldsymbol{x} = \boldsymbol{b}$ changes after changing a row and column of $\mathbf{C}$ or after a rank-r change in $\mathbf{C}$.

Key words. numerical linear algebra, direct methods, Cholesky factorization, sparse matrices, mathematical software, matrix updates

AMS subject classifications. 65F05, 65F50, 65Y20

DOI. 10.1137/S089547980343641X

1. Introduction.

The problem of updating a Cholesky factorization after a small rank change in the matrix is a fundamental problem with many applications, including optimization algorithms, least-squares problems in statistics, the analysis of electrical circuits and power systems, structural mechanics, boundary condition changes in partial differential equations, domain decomposition methods, and boundary element methods (see [12]). Some specific examples follow.

1. A linear programming problem has the form

$$\min\ \boldsymbol{c}^\mathsf{T}\boldsymbol{x}\ \text{ subject to }\ \mathbf{A}\boldsymbol{x} = \boldsymbol{b},\quad \boldsymbol{x} \geq \mathbf{0}, \tag{1.1}$$

where $\mathbf{A}$ is m-by-n, typically n is much larger than m, and all vectors are of compatible size. In this formulation, the vector $\boldsymbol{x}$ is called the primal variable. The dual approach utilizes a multiplier $\boldsymbol{\lambda}$ corresponding to the linear equation $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$. In each iteration of the linear programming dual active set algorithm (LPDASA) (see [5, 13, 14, 15, 16, 17]), we solve a symmetric linear system of the form

$$\mathbf{C}\boldsymbol{\lambda} = \boldsymbol{f},\quad \mathbf{C} = \mathbf{A}_F\mathbf{A}_F^\mathsf{T} + \sigma\mathbf{I},$$

where $\sigma > 0$ is a small parameter, $F \subset \{1, 2, \ldots, n\}$ are the indices associated with "free variables" (strictly positive primal variables), $\mathbf{A}_F$ is a submatrix of $\mathbf{A}$ associated with column indices in F, and $\boldsymbol{f}$ is a function of $\boldsymbol{b}$ and $\boldsymbol{c}$. As the dual iterates converge to optimality, the set F changes as the primal variables either reach their bound or become free. Since $\mathbf{C}$ can be expressed as

$$\mathbf{C} = \sum_{j\in F} \mathbf{A}_{*j}\mathbf{A}_{*j}^\mathsf{T} + \sigma\mathbf{I},$$

*Received by the editors October 21, 2003; accepted for publication (in revised form) by E. Ng April 9, 2004; published electronically March 3, 2005. This material is based upon work supported by the National Science Foundation under grant CCR-0203270.
http://www.siam.org/journals/simax/26-3/43641.html

†Department of Computer and Information Science and Engineering, University of Florida, P.O. Box 116120, Gainesville, FL 32611-6120 (davis@cise.ufl.edu, http://www.cise.ufl.edu/∼davis).

‡Department of Mathematics, University of Florida, P.O. Box 118105, Gainesville, FL 32611-8105 (hager@math.ufl.edu, http://www.math.ufl.edu/∼hager).

where $\mathbf{A}_{*j}$ denotes the jth column of $\mathbf{A}$, it follows that a small change in F leads to a small rank change in $\mathbf{C}$; hence, we solve a sequence of linear systems where each matrix is a small rank modification of the previous matrix.

2. Consider a network of resistors connecting nodes $\{1, 2, \ldots, n\}$ in a graph. Let $\mathcal{A}_i$ denote the set of nodes adjacent to i in the graph, let R_{ij} be the resistance between i and j, and let V_j be the potential at node j (some of the nodes may be held at a fixed potential by a battery). By Kirchhoff's first law, the sum of the currents entering each node is zero:

$$\sum_{j \in \mathcal{A}_i} \frac{V_j - V_i}{R_{ij}} = 0.$$

If the resistance on an arc (k, l) is changed from R_{kl} to $\bar{R}_{kl}$, then there is a rank-1 change in the matrix given by

$$\left(\frac{1}{R_{kl}} - \frac{1}{\bar{R}_{kl}} \right) \boldsymbol{w}\boldsymbol{w}^{\mathsf{T}}, \quad \boldsymbol{w} = \boldsymbol{e}_k - \boldsymbol{e}_l,$$

where $\boldsymbol{e}_i$ is the ith column of the identity matrix. In other words, the only change in the coefficient matrix occurs in rows k and l and in columns k and l. Changing the resistance on r arcs in the network corresponds to a rank-r change in the matrix. Additional illustrations can be found in [12].

A variety of techniques for modifying a dense Cholesky factorization are given in the classic reference [11]. Recently in [3, 4] we considered a sparse Cholesky factorization $\mathbf{LDL}^{\mathsf{T}}$ of a symmetric, positive definite matrix $\mathbf{C}$, and the modification associated with a rank-r change of the form $\overline{\mathbf{C}} = \mathbf{C} \pm \mathbf{W}\mathbf{W}^{\mathsf{T}}$, where $\mathbf{W}$ is n-by-r with r typically much less than n. In a rank-1 update of the form $\overline{\mathbf{C}} = \mathbf{C} + \boldsymbol{w}\boldsymbol{w}^{\mathsf{T}}$, the columns that change in $\mathbf{L}$ correspond to a path in the elimination tree of the modified factor $\overline{\mathbf{L}}$. The path starts at the node corresponding to the row index of the first nonzero entry in $\boldsymbol{w}$. The total work of the rank-1 update is proportional to the number of entries in $\mathbf{L}$ that change, so our algorithm is optimal. A downdate is analogous; it follows a path in the original elimination tree, which becomes a subtree in the new elimination tree.

A rank-r update of the form $\overline{\mathbf{C}} = \mathbf{C} + \mathbf{W}\mathbf{W}^{\mathsf{T}}$, where $\mathbf{W}$ has r columns, can be cast as a sequence of r rank-1 updates. In [4], we show that a rank-r update can be done more efficiently in a single pass. Rather than following a single path in the tree, multiple paths are followed. When paths merge, multiple updates are performed to the corresponding columns of $\mathbf{L}$. Our rank-r algorithm is also optimal.

Figure 1.1 shows an example of a sparse rank-2 update (see Figure 4.1 in [4]). Entries that change in $\overline{\mathbf{C}}$ are shown as a plus. It is not shown in the figure, but the updates follow two paths in the tree, one with nodes $\{1, 2, 6, 8\}$ and the second one with nodes $\{3, 4, 5, 6, 7, 8\}$.

In this paper, we consider a special, but important, rank-2 change corresponding to a symmetric modification of a row and column of $\mathbf{C}$. Although we could, in principle, use our previous methodology to update the factorization, we observe that this rank-2 approach is much less efficient than the streamlined approach we develop here. In fact, the rank-r approach with $r = 2$ could result in a completely dense modification of the factorization, where nonzero entries are first introduced and then canceled out. Figure 1.2 shows a sparse modification to row 4 and column 4 of the matrix $\mathbf{C}$ from Figure 1.1.

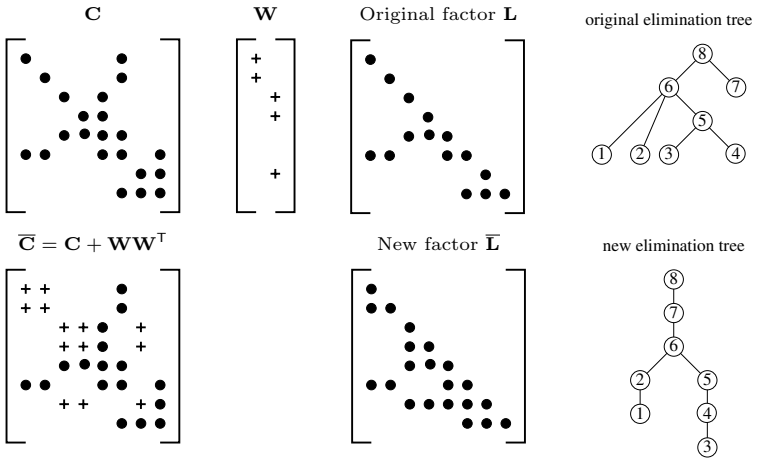


FIG. 1.1. *Rank-2 update,* $\overline{\mathbf{C}} = \mathbf{C} + \mathbf{W}\mathbf{W}^{\mathsf{T}}$.

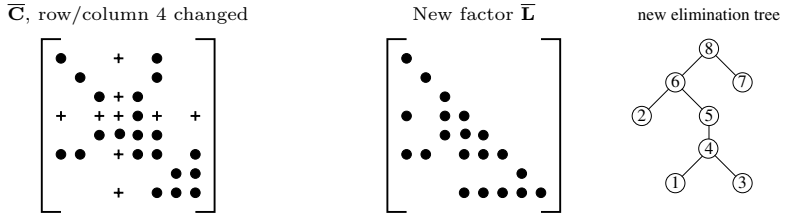


FIG. 1.2. *Modification to a row and column of* $\mathbf{C}$.

With the new approach, the work connected with *removing* a nonzero row and column is comparable to the work associated with a sparse rank-1 *update*, while the work associated with *adding* a new nonzero row and column is comparable to the work associated with a sparse rank-1 *downdate*. This connection between the modification of the matrix and the modification of the factorization is nonintuitive: When we remove elements from the matrix, we update the factorization; when we add elements to the matrix, we downdate the factorization.

As a byproduct of our analysis, we show how the solution to a triangular system $\mathbf{L}\boldsymbol{x} = \boldsymbol{b}$ changes when both $\mathbf{L}$ and $\boldsymbol{b}$ change as a result of the row and column modification problem discussed in this paper, or as a result of a rank-r change to $\mathbf{C}$ [3, 4].

One specific application for the techniques developed in this paper is LPDASA. An inequality $\boldsymbol{a}^{\mathsf{T}}\boldsymbol{x} \leq b$ in a primal linear program is converted to an equality, when the problem is written in the standard form (1.1), by introducing a primal slack variable: $\boldsymbol{a}^{\mathsf{T}}\boldsymbol{x} + y = b$, where $y \geq 0$ is the slack variable. If the index j corresponds to a primal slack variable in equation i, and if $j \in F$, then it can be shown that $\lambda_i = c_j$. In essence, we can eliminate the ith dual variable and the ith equation: The ith equality is satisfied by simply solving for the value of the slack variable, and the ith dual variable is $\lambda_i = c_j$. Thus, in this dual approach to linear programming, inactive inequalities are identified dynamically, during the solution process; dropping these inactive inequalities amounts to removing a row and a column from a Cholesky

factorized matrix. In the same way, when a dropped inequality later becomes active, a new row and column must be inserted in the matrix, and the resulting modification in the Cholesky factorization evaluated. In general, the techniques developed in this paper are useful in any setting where a system of equations is solved repeatedly with equations added or dropped before each solve.

A brief overview of our paper follows. In section 2 we consider the *row addition* problem, in which a row and column, originally zero except for the diagonal element, are modified in a matrix. The *row deletion* problem is the opposite of row addition and is discussed in section 3. Section 4 describes modifications to a row and column of sparse or dense $\mathbf{C}$. We show that arbitrary modifications can be efficiently implemented as a row deletion followed by a row addition. In contrast, we also show that if sparse modifications to a sparse row of $\mathbf{C}$ are made, some improvement can be obtained over a row deletion followed by a row addition. The efficient methods presented in sections 2 through 4 are contrasted with performing the modifications as a rank-2 outer product modification in section 5, which is shown to be costly, particularly in the sparse case. Section 6 shows how to efficiently modify the solution to $\mathbf{L}\boldsymbol{x} = \boldsymbol{b}$ when $\mathbf{L}$ and $\boldsymbol{b}$ change. A brief presentation of the experimental performance of these methods in the context of matrices arising in linear programming is given in section 7.

We use the notation $\overline{\mathbf{C}}$ to denote the matrix $\mathbf{C}$ after it has been modified. Bold uppercase $\mathbf{A}$ refers to a matrix. Bold lowercase italic $\boldsymbol{a}$ is a column vector; thus, $\boldsymbol{a}^{\mathsf{T}}$ always refers to a row vector. Plain lowercase letters (such as a and α) are scalars. We use $|\mathbf{A}|$ to denote the number of nonzero entries in the sparse matrix $\mathbf{A}$. Without parentheses, the notation $\mathbf{A}_i$ or $\mathbf{A}_{ij}$ refers to submatrices of a matrix $\mathbf{A}$ (sometimes 1-by-1 submatrices). We use parentheses $(\mathbf{A})_{ij}$ to refer to the entry in row i and column j of the matrix $\mathbf{A}$, $(\mathbf{A})_{*j}$ to refer to column j of $\mathbf{A}$, and $(\mathbf{A})_{i*}$ to refer to row i of $\mathbf{A}$. When counting floating-point operations (flops), we count one flop for any arithmetic operation including $*$, $/$, $+$, $-$, and $\sqrt{}$.

2. Adding a row and column to C. If we have a rectangular n-by-m matrix $\mathbf{A}$, and $\mathbf{C} = \alpha\mathbf{I} + \mathbf{A}\mathbf{A}^{\mathsf{T}}$, then modifying row k of $\mathbf{A}$ leads to changes in the kth row and column of $\mathbf{C}$. In this section, we consider the special case where row k is initially zero and becomes nonzero in $\overline{\mathbf{A}}$ (the *row addition* case). Equivalently, the kth row and column of $\mathbf{C}$ is initially a multiple of the kth row and column of the identity matrix and changes to some other value.

We first discuss the linear algebra that applies whether $\mathbf{C}$ is dense or sparse. Specific issues for the dense and sparse case are discussed in sections 2.1 and 2.2.

Let $\overline{\boldsymbol{a}}_2^{\mathsf{T}}$ be the new nonzero kth row of $\overline{\mathbf{A}}$,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0}^{\mathsf{T}} \\ \mathbf{A}_3 \end{bmatrix}, \quad \overline{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_1 \\ \overline{\boldsymbol{a}}_2^{\mathsf{T}} \\ \mathbf{A}_3 \end{bmatrix},$$

where $\mathbf{0}^{\mathsf{T}}$ is a row vector whose entries are all 0. This leads to a modification to row and column k of $\mathbf{C}$,

$$\mathbf{C} = \begin{bmatrix} \alpha\mathbf{I} + \mathbf{A}_1\mathbf{A}_1^{\mathsf{T}} & \mathbf{0} & \mathbf{A}_1\mathbf{A}_3^{\mathsf{T}} \\ \mathbf{0}^{\mathsf{T}} & \alpha & \mathbf{0}^{\mathsf{T}} \\ \mathbf{A}_3\mathbf{A}_1^{\mathsf{T}} & \mathbf{0} & \alpha\mathbf{I} + \mathbf{A}_3\mathbf{A}_3^{\mathsf{T}} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} & \mathbf{C}_{31}^{\mathsf{T}} \\ \mathbf{0}^{\mathsf{T}} & c_{22} & \mathbf{0}^{\mathsf{T}} \\ \mathbf{C}_{31} & \mathbf{0} & \mathbf{C}_{33} \end{bmatrix},$$

where $c_{22} = \alpha$. We can let α be zero; even though $\mathbf{C}$ would no longer be positive definite, the submatrix excluding row and column k could still be positive definite.

The linear system $\mathbf{C}\boldsymbol{x} = \boldsymbol{b}$ is well defined in this case, except for $\boldsymbol{x}_k$. The new matrix $\overline{\mathbf{C}}$ is given as

$$\overline{\mathbf{C}} = \begin{bmatrix} \alpha\mathbf{I} + \mathbf{A}_1\mathbf{A}_1^\mathsf{T} & \mathbf{A}_1\overline{\boldsymbol{a}}_2 & \mathbf{A}_1\mathbf{A}_3^\mathsf{T} \\ \overline{\boldsymbol{a}}_2^\mathsf{T}\mathbf{A}_1^\mathsf{T} & \alpha + \overline{\boldsymbol{a}}_2^\mathsf{T}\overline{\boldsymbol{a}}_2 & \overline{\boldsymbol{a}}_2^\mathsf{T}\mathbf{A}_3^\mathsf{T} \\ \mathbf{A}_3\mathbf{A}_1^\mathsf{T} & \mathbf{A}_3\overline{\boldsymbol{a}}_2 & \alpha\mathbf{I} + \mathbf{A}_3\mathbf{A}_3^\mathsf{T} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{11} & \overline{\boldsymbol{c}}_{12} & \mathbf{C}_{31}^\mathsf{T} \\ \overline{\boldsymbol{c}}_{12}^\mathsf{T} & \overline{c}_{22} & \overline{\boldsymbol{c}}_{32}^\mathsf{T} \\ \mathbf{C}_{31} & \overline{\boldsymbol{c}}_{32} & \mathbf{C}_{33} \end{bmatrix}.$$

Thus, adding a row k to $\mathbf{A}$ is equivalent to adding a row and column k to $\mathbf{C}$. Note that changing row and column k of $\mathbf{C}$ from zero (except for the diagonal entry) to a nonzero value also can be viewed as increasing the dimension of the $(n-1)$-by-$(n-1)$ matrix

$$\begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{31}^\mathsf{T} \\ \mathbf{C}_{31} & \mathbf{C}_{33} \end{bmatrix}.$$

The original factorization of the n-by-n matrix $\mathbf{C}$ may be written as

$$\begin{aligned} \mathbf{L}\mathbf{D}\mathbf{L}^\mathsf{T} &= \begin{bmatrix} \mathbf{L}_{11} & & \\ \mathbf{0}^\mathsf{T} & 1 & \\ \mathbf{L}_{31} & \mathbf{0} & \mathbf{L}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{D}_{11} & & \\ & d_{22} & \\ & & \mathbf{D}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{L}_{11}^\mathsf{T} & \mathbf{0} & \mathbf{L}_{31}^\mathsf{T} \\ & 1 & \mathbf{0}^\mathsf{T} \\ & & \mathbf{L}_{33}^\mathsf{T} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} & \mathbf{C}_{31}^\mathsf{T} \\ \mathbf{0}^\mathsf{T} & \alpha & \mathbf{0}^\mathsf{T} \\ \mathbf{C}_{31} & \mathbf{0} & \mathbf{C}_{33} \end{bmatrix}, \end{aligned}$$

which leads to the four equations

$$\begin{aligned} \mathbf{L}_{11}\mathbf{D}_{11}\mathbf{L}_{11}^\mathsf{T} &= \mathbf{C}_{11}, \\ d_{22} &= \alpha, \\ \mathbf{L}_{31}\mathbf{D}_{11}\mathbf{L}_{11}^\mathsf{T} &= \mathbf{C}_{31}, \\ \mathbf{L}_{31}\mathbf{D}_{11}\mathbf{L}_{31}^\mathsf{T} + \mathbf{L}_{33}\mathbf{D}_{33}\mathbf{L}_{33}^\mathsf{T} &= \mathbf{C}_{33}. \end{aligned} \tag{2.1}$$

After adding row and column k to obtain $\overline{\mathbf{C}}$, we have the factorization

$$\begin{aligned} \overline{\mathbf{L}\mathbf{D}\mathbf{L}}^\mathsf{T} &= \begin{bmatrix} \mathbf{L}_{11} & & \\ \bar{\boldsymbol{l}}_{12}^\mathsf{T} & 1 & \\ \mathbf{L}_{31} & \bar{\boldsymbol{l}}_{32} & \overline{\mathbf{L}}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{D}_{11} & & \\ & \overline{d}_{22} & \\ & & \overline{\mathbf{D}}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{L}_{11}^\mathsf{T} & \bar{\boldsymbol{l}}_{12} & \mathbf{L}_{31}^\mathsf{T} \\ & 1 & \bar{\boldsymbol{l}}_{32}^\mathsf{T} \\ & & \overline{\mathbf{L}}_{33}^\mathsf{T} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{C}_{11} & \overline{\boldsymbol{c}}_{12} & \mathbf{C}_{31}^\mathsf{T} \\ \overline{\boldsymbol{c}}_{12}^\mathsf{T} & \overline{c}_{22} & \overline{\boldsymbol{c}}_{32}^\mathsf{T} \\ \mathbf{C}_{31} & \overline{\boldsymbol{c}}_{32} & \mathbf{C}_{33} \end{bmatrix}. \end{aligned} \tag{2.2}$$

Note that $\mathbf{L}_{11}$ and $\mathbf{L}_{31}$ do not change as a result of modifying row and column k of $\mathbf{C}$. From (2.2), the relevant equations are

$$\mathbf{L}_{11}\mathbf{D}_{11}\bar{\boldsymbol{l}}_{12} = \overline{\boldsymbol{c}}_{12}, \tag{2.3}$$

$$\begin{aligned} \bar{\boldsymbol{l}}_{12}^\mathsf{T}\mathbf{D}_{11}\bar{\boldsymbol{l}}_{12} + \overline{d}_{22} &= \overline{c}_{22}, \\ \mathbf{L}_{31}\mathbf{D}_{11}\bar{\boldsymbol{l}}_{12} + \bar{\boldsymbol{l}}_{32}\overline{d}_{22} &= \overline{\boldsymbol{c}}_{32}, \\ \mathbf{L}_{31}\mathbf{D}_{11}\mathbf{L}_{31}^\mathsf{T} + \bar{\boldsymbol{l}}_{32}\overline{d}_{22}\bar{\boldsymbol{l}}_{32}^\mathsf{T} + \overline{\mathbf{L}}_{33}\overline{\mathbf{D}}_{33}\overline{\mathbf{L}}_{33}^\mathsf{T} &= \mathbf{C}_{33}. \end{aligned} \tag{2.4}$$

Let $\boldsymbol{w} = \bar{\boldsymbol{l}}_{32}\sqrt{\bar{d}_{22}}$. Combining (2.4) with the original equation (2.1), we obtain

$$\overline{\mathbf{L}}_{33}\overline{\mathbf{D}}_{33}\overline{\mathbf{L}}_{33}^{\mathsf{T}} = (\mathbf{C}_{33} - \mathbf{L}_{31}\mathbf{D}_{11}\mathbf{L}_{31}^{\mathsf{T}}) - \bar{\boldsymbol{l}}_{32}\bar{d}_{22}\bar{\boldsymbol{l}}_{32}^{\mathsf{T}} = \mathbf{L}_{33}\mathbf{D}_{33}\mathbf{L}_{33}^{\mathsf{T}} - \boldsymbol{w}\boldsymbol{w}^{\mathsf{T}}.$$

The factorization of $\mathbf{L}_{33}\mathbf{D}_{33}\mathbf{L}_{33}^{\mathsf{T}} - \boldsymbol{w}\boldsymbol{w}^{\mathsf{T}}$ can be computed as a rank-1 downdate of the original factorization $\mathbf{L}_{33}\mathbf{D}_{33}\mathbf{L}_{33}^{\mathsf{T}}$ (see [3]). This derivation leads to Algorithm 1 for computing the modified factorization $\overline{\mathbf{L}\mathbf{D}\mathbf{L}}^{\mathsf{T}}$, which is applicable in both the dense and sparse cases.

ALGORITHM 1 (ROW ADDITION).

1. Solve the lower triangular system $\mathbf{L}_{11}\mathbf{D}_{11}\bar{\boldsymbol{l}}_{12} = \bar{\boldsymbol{c}}_{12}$ for $\bar{\boldsymbol{l}}_{12}$.
2. $\bar{d}_{22} = \bar{c}_{22} - \bar{\boldsymbol{l}}_{12}^{\mathsf{T}}\mathbf{D}_{11}\bar{\boldsymbol{l}}_{12}$
3. $\bar{\boldsymbol{l}}_{32} = (\bar{\boldsymbol{c}}_{32} - \mathbf{L}_{31}\mathbf{D}_{11}\bar{\boldsymbol{l}}_{12})/\bar{d}_{22}$
4. $\boldsymbol{w} = \bar{\boldsymbol{l}}_{32}\sqrt{\bar{d}_{22}}$
5. Perform the rank-1 downdate $\overline{\mathbf{L}}_{33}\overline{\mathbf{D}}_{33}\overline{\mathbf{L}}_{33}^{\mathsf{T}} = \mathbf{L}_{33}\mathbf{D}_{33}\mathbf{L}_{33}^{\mathsf{T}} - \boldsymbol{w}\boldsymbol{w}^{\mathsf{T}}$.

end Algorithm 1

2.1. Dense row addition. Consider the case when $\mathbf{C}$ is dense.

1. Step (1) of Algorithm 1 requires the solution of a unit lower triangular system $\mathbf{L}_{11}\boldsymbol{y} = \bar{\boldsymbol{c}}_{12}$ of order $k-1$. The computation of $\boldsymbol{y}$ takes $(k-1)^2 - (k-1)$ flops, and computing $\bar{\boldsymbol{l}}_{12} = \mathbf{D}_{11}^{-1}\boldsymbol{y}$ takes another $k-1$ flops.
2. Step (2) requires $2(k-1)$ work, using $\boldsymbol{y}$.
3. Step (3) is the matrix-vector multiply $\mathbf{L}_{31}\boldsymbol{y}$, where $\mathbf{L}_{31}$ is $(n-k)$-by-$(k-1)$ and thus takes $2(n-k)(k-1)$ operations and $k-1$ more to divide by $\bar{d}_{22}$.
4. Step (4) requires one square root operation and $n-k$ multiplications.
5. Finally, the rank-1 downdate of step (5) takes $2(n-k)^2 + 4(n-k)$ operations using method C1 of [11] (see [4]).

For the dense case, the total number of flops performed by Algorithm 1 is $2n^2 + 3n + k^2 - (2nk + 2k + 1)$. This is roughly $2n^2$ when $k = 1$, n^2 when $k = n$, and $(5/4)n^2$ when $k = n/2$.

2.2. Sparse row addition. If $\mathbf{C}$ is sparse, each step of Algorithm 1 must operate on sparse matrices. The graph algorithms and data structures must efficiently support each step. We will assume that $\mathbf{L}$ is stored in a compressed column vector form, where the row indices in each column are sorted in ascending order. This is the same data structure used in [3, 4], except that the algorithms presented there do not require sorted row indices, but they do require the integer *multiplicity* of each nonzero entry of $\mathbf{L}$ to support an efficient symbolic downdate operation. The algorithm discussed below will not require the multiplicities.

Maintaining the row indices in sorted order requires a merge operation for the set union computation to determine the new nonzero patterns of the columns of $\mathbf{L}$, rather than a simpler unsorted set union used in [3, 4]. It has no effect on asymptotic complexity and little effect on the run time. Although more work is required to maintain the row indices in sorted order, time is gained elsewhere in the algorithm. Operating on columns in sorted order in the forward solve of $\mathbf{L}\boldsymbol{x} = \boldsymbol{b}$, for example, is faster than operating on a matrix with jumbled columns. No additional space is required to keep the columns sorted.

Step (1) of Algorithm 1 solves the lower triangular system $\mathbf{L}_{11}\boldsymbol{y} = \bar{\boldsymbol{c}}_{12}$, where all three terms in this system are sparse. Gilbert and Peierls have shown how to solve this system optimally, in time proportional to the number of flops required [9, 10]. We review their method here.

Consider the n-by-n system $\mathbf{L}\boldsymbol{x} = \boldsymbol{b}$, where $\mathbf{L}$ is lower triangular and both $\mathbf{L}$ and $\boldsymbol{b}$ are sparse. The solution will be sparse, and the total work may be less than $O(n)$. We cannot use a conventional algorithm that iterates over each column of $\mathbf{L}$ and skips those for which $\boldsymbol{x}$ is zero since the work involved will then be at least n. Instead, the nonzero pattern of $\boldsymbol{x}$ must first be computed, and then the corresponding columns of $\mathbf{L}$ can be used to compute $\boldsymbol{x}$ in time proportional to the floating-point work required.

Let G_L be a graph with n nodes and with a directed edge from node j to node i if and only if l_{ij} is nonzero. Gilbert and Peierls show that x_j is nonzero (ignoring numerical cancellation) if and only if there is a path of length zero or more from some node i, where $b_i \neq 0$, to node j in the graph G_L. Computing the pattern of $\boldsymbol{x}$ requires a graph traversal, starting from the nodes corresponding to the nonzero pattern of $\boldsymbol{b}$. It can be done in time proportional to the number of edges traversed. Each of these edges is a nonzero in $\mathbf{L}$ that takes part in the subsequent numerical computation.

The above result holds for any lower triangular matrix $\mathbf{L}$. In our case, $\mathbf{L}$ arises from a Cholesky factorization and has an *elimination tree* [18, 19]. The elimination tree of $\mathbf{L}$ has n nodes. The parent of node j in the elimination tree is the smallest index $i > j$ for which $l_{ij} \neq 0$; node j is a root if there is no such i. Since the nonzero pattern of $(\mathbf{L})_{*j}$ is a subset of its path to the root of the elimination tree [20], all the nodes in G_L that can be reached from node j correspond to the path from node j to the root of the elimination tree. Traversing the paths in the tree, starting at nodes corresponding to nonzero entries in $\boldsymbol{b}$, takes time proportional to the number of nonzero entries in $\boldsymbol{x}$. A general graph traversal of G_L is not required. Step (1) of Algorithm 1 takes

$$O\left(\sum_{(\bar{\boldsymbol{l}}_{12})_j \neq 0} |(\mathbf{L}_{11})_{*j}|\right)$$

time to compute the nonzero pattern and numerical values of both $\boldsymbol{y}$ and $\bar{\boldsymbol{l}}_{12}$. The insertion of the nonzero entries of $\bar{\boldsymbol{l}}_{12}$ into the data structure of $\mathbf{L}$ is performed in conjunction with step (3).

Step (2) is a scaled dot product operation and can be computed in time proportional to the number of nonzero entries in $\bar{\boldsymbol{l}}_{12}$.

Step (3) is a matrix-vector multiply operation. It accesses the same columns of $\mathbf{L}$ used by the sparse lower triangular solve, namely, each column j for which the jth entry in $\bar{\boldsymbol{l}}_{12}$ is nonzero. These same columns need to be modified by shifting entries in $\mathbf{L}_{31}$ down by one and inserting the new entries in $\bar{\boldsymbol{l}}_{12}$, the kth row of $\overline{\mathbf{L}}$. No other columns in the range 1 to $k-1$ need to be accessed or modified by steps (1) through (3). When step (3) completes, the new column k of $\overline{\mathbf{L}}$ needs to be inserted into the data structure. This can be done in one of two ways. In the general case, we can store the columns themselves in a noncontiguous manner and simply allocate new space for this column. A similar strategy can be used for any columns 1 through $k-1$ of $\overline{\mathbf{L}}$ that outgrow their originally allocated space with no increase in asymptotic run time. Alternatively, we may know an a priori upper bound on the size of each column of $\mathbf{L}$ after all row additions have been performed. In this case, a simpler static allocation strategy is possible. This latter case occurs in our use of the row addition algorithm in LPDASA, our target application [5]. In either case, the time to insert $\bar{\boldsymbol{l}}_{12}$ into the data structure and to compute $\bar{\boldsymbol{l}}_{32}$ in step (3) is

$$O\left(\sum_{(\bar{\boldsymbol{l}}_{12})_j \neq 0} |(\mathbf{L}_{31})_{*j}|\right).$$

Step (4) is a simple scalar-times-vector operation. The total time for steps (1) through (4) is

$$O\left(\sum_{(\bar{\boldsymbol{l}}_{12})_j \neq 0} |(\mathbf{L})_{*j}|\right).$$

Step (5) almost fits the specifications of the sparse rank-1 modification in [3], but with one interesting twist. The original kth row and column of $\mathbf{C}$ are zero, except for the placeholder diagonal entry, α. The new row and column only add entries to $\mathbf{C}$, and thus the nonzero pattern of the original factor $\mathbf{L}$ is a subset of the nonzero pattern of $\overline{\mathbf{L}}$ (ignoring numerical cancellation). The rank-1 modification in Algorithm 1 is a symbolic update (new nonzero entries are added, not removed) and a numeric downdate $\mathbf{L}_{33}\mathbf{D}_{33}\mathbf{L}_{33}^{\mathsf{T}} - \boldsymbol{w}\boldsymbol{w}^{\mathsf{T}}$. Since the multiplicities used in [3] are needed only for a subsequent symbolic downdate, they are not required by the row addition algorithm. They would be required by a row deletion algorithm that maintains a strict nonzero pattern of $\mathbf{L}$; this issue is addressed in section 3.2.

The rank-1 modification to obtain the factorization $\overline{\mathbf{L}}_{33}\overline{\mathbf{D}}_{33}\overline{\mathbf{L}}_{33}^{\mathsf{T}}$ takes time proportional to the number of nonzero entries in $\overline{\mathbf{L}}_{33}$ that change. The columns that change correspond to the path from node k to the root of the elimination tree of $\overline{\mathbf{L}}$. This path is denoted $\overline{\mathcal{P}}$ in [3]. At each node j along the path $\overline{\mathcal{P}}$, at most four flops are performed for each nonzero entry in $(\overline{\mathbf{L}})_{*j}$.

With our choice of data structures, exploitation of the elimination tree, and the rank-1 modification from [3], the total time taken by Algorithm 1 is proportional to the total number of nonzero entries in columns corresponding to nonzero entries in $\bar{\boldsymbol{l}}_{12}$, to compute steps (1) through (4), plus the time required for the rank-1 modification in step (5). The total time is

$$O\left(\sum_{(\bar{\boldsymbol{l}}_{12})_j \neq 0} |(\mathbf{L})_{*j}| + \sum_{j \in \overline{\mathcal{P}}} |(\overline{\mathbf{L}})_{*j}|\right).$$

This time includes all data structure manipulations, sparsity pattern computation, and graph algorithms required to implement the algorithm. It is identical to the total number of flops required, and thus Algorithm 1 is optimal. In the sparse case, if every column j takes part in the computation, the time is $O(|\overline{\mathbf{L}}|)$. Normally, not all columns will be affected by the sparse row addition. If the new row and column of $\overline{\mathbf{L}}$ are very sparse, only a few columns take part in the computation.

3. Row deletion. By *deleting* a row and column k from the matrix $\mathbf{C}$, we mean setting the entire row and column to zero, except for the diagonal entry $(\mathbf{C})_{kk}$ which is set to α. This is the opposite of row addition. Here, we present an algorithm that applies whether $\mathbf{C}$ is sparse or dense. Specific issues in the dense case are considered in section 3.1, and the sparse case is discussed in section 3.2.

Prior to deleting row and column k, we have the original factorization

$$\begin{aligned}\mathbf{L}\mathbf{D}\mathbf{L}^{\mathsf{T}} &= \begin{bmatrix} \mathbf{L}_{11} & & \\ \boldsymbol{l}_{12}^{\mathsf{T}} & 1 & \\ \mathbf{L}_{31} & \boldsymbol{l}_{32} & \mathbf{L}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{D}_{11} & & \\ & d_{22} & \\ & & \mathbf{D}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{L}_{11}^{\mathsf{T}} & \boldsymbol{l}_{12} & \mathbf{L}_{31}^{\mathsf{T}} \\ & 1 & \boldsymbol{l}_{32}^{\mathsf{T}} \\ & & \mathbf{L}_{33}^{\mathsf{T}} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{C}_{11} & \boldsymbol{c}_{12} & \mathbf{C}_{31}^{\mathsf{T}} \\ \boldsymbol{c}_{12}^{\mathsf{T}} & c_{22} & \boldsymbol{c}_{32}^{\mathsf{T}} \\ \mathbf{C}_{31} & \boldsymbol{c}_{32} & \mathbf{C}_{33} \end{bmatrix}.\end{aligned}$$

After deleting row and column k, we have

$$\overline{\mathbf{L}}\overline{\mathbf{D}}\overline{\mathbf{L}}^{\mathsf{T}} = \begin{bmatrix} \mathbf{L}_{11} & & \\ \mathbf{0}^{\mathsf{T}} & 1 & \\ \mathbf{L}_{31} & \mathbf{0} & \overline{\mathbf{L}}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{D}_{11} & & \\ & \alpha & \\ & & \overline{\mathbf{D}}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{L}_{11}^{\mathsf{T}} & \mathbf{0} & \mathbf{L}_{31}^{\mathsf{T}} \\ & 1 & \mathbf{0}^{\mathsf{T}} \\ & & \overline{\mathbf{L}}_{33}^{\mathsf{T}} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} & \mathbf{C}_{31}^{\mathsf{T}} \\ \mathbf{0}^{\mathsf{T}} & \alpha & \mathbf{0}^{\mathsf{T}} \\ \mathbf{C}_{31} & \mathbf{0} & \mathbf{C}_{33} \end{bmatrix}.$$

Thus we only need to set row and column k of $\overline{\mathbf{L}}$ to zero, set the diagonal entry to α, and compute $\overline{\mathbf{L}}_{33}$ and $\overline{\mathbf{D}}_{33}$. The original factorization is

$$\mathbf{L}_{33}\mathbf{D}_{33}\mathbf{L}_{33}^{\mathsf{T}} = \mathbf{C}_{33} - \mathbf{L}_{31}\mathbf{D}_{11}\mathbf{L}_{31}^{\mathsf{T}} - \boldsymbol{l}_{32}d_{22}\boldsymbol{l}_{32}^{\mathsf{T}},$$

while the new factorization is given as

$$\overline{\mathbf{L}}_{33}\overline{\mathbf{D}}_{33}\overline{\mathbf{L}}_{33}^{\mathsf{T}} = \mathbf{C}_{33} - \mathbf{L}_{31}\mathbf{D}_{11}\mathbf{L}_{31}^{\mathsf{T}}.$$

Combining these two equations, we have a numeric rank-1 update,

$$\overline{\mathbf{L}}_{33}\overline{\mathbf{D}}_{33}\overline{\mathbf{L}}_{33}^{\mathsf{T}} = \mathbf{L}_{33}\mathbf{D}_{33}\mathbf{L}_{33}^{\mathsf{T}} + \boldsymbol{w}\boldsymbol{w}^{\mathsf{T}}, \tag{3.1}$$

where $\boldsymbol{w} = \boldsymbol{l}_{32}\sqrt{d_{22}}$. Algorithm 2 gives the complete row deletion algorithm, which is applicable in both the dense and sparse cases.

ALGORITHM 2 (ROW DELETION).

1. $\overline{\boldsymbol{l}}_{12} = \mathbf{0}$
2. $\overline{d}_{22} = \alpha$
3. $\overline{\boldsymbol{l}}_{32} = \mathbf{0}$
4. $\boldsymbol{w} = \boldsymbol{l}_{32}\sqrt{d_{22}}$
5. Perform the rank-1 update $\overline{\mathbf{L}}_{33}\overline{\mathbf{D}}_{33}\overline{\mathbf{L}}_{33}^{\mathsf{T}} = \mathbf{L}_{33}\mathbf{D}_{33}\mathbf{L}_{33}^{\mathsf{T}} + \boldsymbol{w}\boldsymbol{w}^{\mathsf{T}}$.

end Algorithm 2

3.1. Dense row deletion. When $\mathbf{C}$ is dense, the number of flops performed by Algorithm 2 is $2(n-k)^2+5(n-k)+1$ (the same as steps (4) and (5) of Algorithm 1). This is roughly $2n^2$ when $k = 1$, and $(1/2)n^2$ when $k = n/2$. No work is required when $k = n$.

3.2. Sparse row deletion. When row and column k of a sparse $\mathbf{C}$ are deleted to obtain $\overline{\mathbf{C}}$, no new nonzero terms will appear in $\overline{\mathbf{L}}$, and some nonzero entries in $\mathbf{L}$ may become zero. We refer to the deletion of entries in $\mathbf{L}$ as a symbolic downdate [3]. The symbolic downdate is combined with a numeric rank-1 update because of the addition of $\boldsymbol{w}\boldsymbol{w}^{\mathsf{T}}$ in (3.1).

We cannot simply delete entries from $\overline{\mathbf{L}}$ that become numerically zero. An entry in $\overline{\mathbf{L}}$ can be removed only if it becomes *symbolically zero* (that is, its value is zero regardless of the assignment of numerical values to the nonzero pattern of $\mathbf{C}$). If entries are zero because of exact numerical cancellation and are dropped from the data structure of $\mathbf{L}$, then the elimination tree no longer characterizes the structure of the matrix. If the elimination tree is no longer valid, subsequent updates and downdates will not be able to determine which columns of $\mathbf{L}$ must be modified.

Steps (1) through (3) of Algorithm 2 require no numerical work, but they do require some data structure modifications. All of the entries in row and column k

of $\overline{\mathbf{L}}$ become symbolically zero and can be removed. If $\mathbf{L}$ is stored by columns, it is trivial in step (3) to immediately delete all entries in column $\boldsymbol{l}_{32}$. On the other hand, the statement $\bar{\boldsymbol{l}}_{12} = \mathbf{0}$ in step (1) is less obvious. Each nonzero element in row k lies in a different column. We must either set these values to zero, delete them from the data structure, or flag row k as zero and require any subsequent algorithm that accesses the matrix $\mathbf{L}$ to ignored flagged rows. The latter option would lead to a sparse row deletion algorithm with optimal run time but complicates all other algorithms in our application and increases their run time. We choose to search for the row k entries in each column in which they appear and delete them from the data structure.

To set $\boldsymbol{l}_{12}$ to zero and delete the entries from the data structure for $\mathbf{L}$ requires a scan of all the columns j of $\mathbf{L}$ for which the jth entry of $\boldsymbol{l}_{12}$ is nonzero. The time taken for this operation is asymptotically bounded by the time taken for steps (1) through (3) of sparse row addition (Algorithm 1), but the bound is not tight. The nonzero pattern of row k of $\mathbf{L}$ can easily be found from the elimination tree. Finding the row index k in the columns takes less time than step (1) of Algorithm 1 since a binary search can be used. Deleting the entries takes time equivalent to step (3) of Algorithm 1 since we maintain each column with sorted row indices.

The immediate removal of entries in $\overline{\mathbf{L}}_{33}$ can be done using the symbolic rank-1 downdate presented in [3]. However, this requires an additional array of multiplicities, which is one additional integer value for each nonzero in the matrix. Instead, we can allow these entries to become numerically zero (or very small values due to numerical roundoff) and not remove them immediately. Since they become numerically zero (or tiny), they can simply remain in the data structure for the matrix and have no effect on subsequent operations that use the matrix $\mathbf{L}$. The entries can be pruned later on by a complete symbolic factorization, taking $O(|\mathbf{L}|)$ time [6, 7, 8]. If this is done rarely, the overall run time of the application that uses the sparse row deletion algorithm will not be affected adversely.

The asymptotic run time of our sparse row deletion algorithm is the same as sparse row addition (or less, because of the binary search), even though sparse row addition requires more numerical work. This is nonoptimal but no worse than sparse row addition, whose run time is optimal.

4. Row modification. It is possible to generalize our row deletion and addition algorithms to handle the case where the kth row and column of $\mathbf{C}$ is neither originally zero (the row addition case) nor set to zero (the row deletion case), but is changed arbitrarily. Any change of this form can be handled as a row deletion followed by a row addition, but the question may remain as to whether or not it can be done faster as a single step. Here, we show that an arbitrary row modification can be efficiently implemented as a row deletion followed by a row addition. If the changes to the row are sparse, however, some work can be saved by combining the two steps.

4.1. Arbitrary row modification. In this section we show that no flops are saved in a single-pass row modification algorithm, as compared to the row deletion + row addition approach, if the change in the kth row and column is arbitrary. This is true whether $\mathbf{C}$ is sparse or dense.

The original matrix $\mathbf{C}$ and the new matrix $\overline{\mathbf{C}}$ are

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & \boldsymbol{c}_{12} & \mathbf{C}_{31}^{\mathsf{T}} \\ \boldsymbol{c}_{12}^{\mathsf{T}} & c_{22} & \boldsymbol{c}_{32}^{\mathsf{T}} \\ \mathbf{C}_{31} & \boldsymbol{c}_{32} & \mathbf{C}_{33} \end{bmatrix} \quad \text{and} \quad \overline{\mathbf{C}} = \begin{bmatrix} \mathbf{C}_{11} & \bar{\boldsymbol{c}}_{12} & \mathbf{C}_{31}^{\mathsf{T}} \\ \bar{\boldsymbol{c}}_{12}^{\mathsf{T}} & \bar{c}_{22} & \bar{\boldsymbol{c}}_{32}^{\mathsf{T}} \\ \mathbf{C}_{31} & \bar{\boldsymbol{c}}_{32} & \mathbf{C}_{33} \end{bmatrix}. \tag{4.1}$$

Computing $\bar{\boldsymbol{l}}_{12}$, $\bar{d}_{22}$, and $\bar{\boldsymbol{l}}_{32}$ is the same as the row addition algorithm and takes exactly the same number of flops. The original factorization of the (33)-block is

$$\mathbf{L}_{33}\mathbf{D}_{33}\mathbf{L}_{33}^{\mathsf{T}} = \mathbf{C}_{33} - \mathbf{L}_{31}\mathbf{D}_{11}\mathbf{L}_{31}^{\mathsf{T}} - \boldsymbol{l}_{32}d_{22}\boldsymbol{l}_{32}^{\mathsf{T}},$$

while the new factorization is

$$\overline{\mathbf{L}}_{33}\overline{\mathbf{D}}_{33}\overline{\mathbf{L}}_{33}^{\mathsf{T}} = \mathbf{C}_{33} - \mathbf{L}_{31}\mathbf{D}_{11}\mathbf{L}_{31}^{\mathsf{T}} - \bar{\boldsymbol{l}}_{32}\bar{d}_{22}\bar{\boldsymbol{l}}_{32}^{\mathsf{T}}.$$

These can be combined into a rank-1 update and rank-1 downdate

$$\overline{\mathbf{L}}_{33}\overline{\mathbf{D}}_{33}\overline{\mathbf{L}}_{33}^{\mathsf{T}} = \mathbf{L}_{33}\mathbf{D}_{33}\mathbf{L}_{33}^{\mathsf{T}} + \boldsymbol{w}_1\boldsymbol{w}_1^{\mathsf{T}} - \boldsymbol{w}_2\boldsymbol{w}_2^{\mathsf{T}}, \tag{4.2}$$

where $\boldsymbol{w}_1 = \boldsymbol{l}_{32}\sqrt{d_{22}}$ and $\boldsymbol{w}_2 = \bar{\boldsymbol{l}}_{32}\sqrt{\bar{d}_{22}}$. The multiple rank update/downdate presented in [4] cannot perform a simultaneous update and downdate, but if the sparse downdate (removal of entries that become symbolically zero) is not performed, it would be possible to compute the simultaneous update and downdate (4.2) in a single pass. This may result in some time savings since the data structure for $\mathbf{L}$ is scanned once, not twice. No floating-point work would be saved, however. The total flop count of the row modification algorithm is identical to a row deletion followed by a row addition.

4.2. Sparse row modification. Suppose we modify some, but not all, of the elements of the kth row and column of $\mathbf{C}$. In this case, we can reduce the total amount of floating-point work required to compute the modified factorization $\overline{\mathbf{L}\mathbf{D}\mathbf{L}}^{\mathsf{T}}$, as shown below.

More precisely, consider the case where only a few entries in row and column k of $\mathbf{C}$ are changed. Let

$$\begin{aligned}\Delta\boldsymbol{c}_{12} &= \bar{\boldsymbol{c}}_{12} - \boldsymbol{c}_{12},\\ \Delta c_{22} &= \bar{c}_{22} - c_{22},\\ \Delta\boldsymbol{c}_{32} &= \bar{\boldsymbol{c}}_{32} - \boldsymbol{c}_{32},\end{aligned}$$

and assume that the change in the kth row and column is much sparser than in the original kth row and column of $\mathbf{C}$ (for example, $|\Delta\boldsymbol{c}_{12}| \ll |\bar{\boldsymbol{c}}_{12}|$).

If we consider (2.3) and its analog for the original matrix $\mathbf{C}$, we have

$$\mathbf{L}_{11}\mathbf{D}_{11}\boldsymbol{l}_{12} = \boldsymbol{c}_{12}, \tag{4.3}$$

$$\mathbf{L}_{11}\mathbf{D}_{11}\bar{\boldsymbol{l}}_{12} = \bar{\boldsymbol{c}}_{12}. \tag{4.4}$$

Combining these two equations gives

$$\mathbf{L}_{11}\mathbf{D}_{11}\Delta\boldsymbol{l}_{12} = \Delta\boldsymbol{c}_{12},$$

where $\Delta\boldsymbol{l}_{12} = \bar{\boldsymbol{l}}_{12} - \boldsymbol{l}_{12}$. Since $\Delta\boldsymbol{c}_{12}$ is sparse, the solution of this lower triangular system will be sparse in general. It can be solved in time proportional to the time required to multiply $\mathbf{L}_{11}$ times $\Delta\boldsymbol{l}_{12}$, or

$$O\left(\sum_{(\Delta\boldsymbol{l}_{12})_j \neq 0} |(\mathbf{L}_{11})_{*j}|\right)$$

[9, 10]. We can then compute $\bar{\boldsymbol{l}}_{12} = \boldsymbol{l}_{12} + \Delta\boldsymbol{l}_{12}$. This approach for computing $\bar{\boldsymbol{l}}_{12}$ can be much faster than solving (4.4) directly, which would take

$$O\left(\sum_{(\bar{\boldsymbol{l}}_{12})_j \neq 0} |(\mathbf{L}_{11})_{*j}|\right)$$

time. Computing $\bar{d}_{22}$ can be done in time proportional to $|\Delta\boldsymbol{l}_{12}|$, using the following formula that modifies the dot product computation in Algorithm 1:

$$\bar{d}_{22} = d_{22} + \Delta d_{22} = d_{22} + \Delta c_{22} - \sum_{(\Delta\boldsymbol{l}_{12})_j \neq 0} (\Delta\boldsymbol{l}_{12})_j((\boldsymbol{l}_{12})_j + (\bar{\boldsymbol{l}}_{12})_j)(\mathbf{D}_{11})_{jj}.$$

Similarly, the kth column of $\overline{\mathbf{L}}$ can be computed as

$$\bar{\boldsymbol{l}}_{32} = \left(\Delta\mathbf{C}_{32} + \boldsymbol{l}_{32}d_{22} - \mathbf{L}_{31}\mathbf{D}_{11}\Delta\boldsymbol{l}_{12}\right)/\bar{d}_{22}.$$

The key component in this computation is the sparse matrix-vector multiplication $\mathbf{L}_{31}\mathbf{D}_{11}\Delta\boldsymbol{l}_{12}$, which takes less time to compute than the corresponding computation $\mathbf{L}_{31}\mathbf{D}_{11}\boldsymbol{l}_{12}$ in step (2) of Algorithm 1.

The remaining work for the rank-1 update/downdate of $\mathbf{L}_{33}\mathbf{D}_{33}\mathbf{L}_{33}^{\mathsf{T}}$ is identical to the arbitrary row modification (4.2). If k is small, it is likely that this update/downdate computation will take much more time than the computation of $\bar{\boldsymbol{l}}_{12}$, d_{22}, and $\bar{\boldsymbol{l}}_{32}$. If k is large, however, a significant reduction in the total amount of work can be obtained by exploiting sparsity in the change in the row and column of $\mathbf{C}$.

5. Row modifications as a rank-2 outer-product. Modifying row and column k of a symmetric matrix $\mathbf{C}$ can be written as a rank-2 modification $\overline{\mathbf{C}} = \mathbf{C} + \boldsymbol{w}_1\boldsymbol{w}_1^{\mathsf{T}} - \boldsymbol{w}_2\boldsymbol{w}_2^{\mathsf{T}}$. Suppose $\mathbf{C}$ and $\overline{\mathbf{C}}$ are as given in (4.1). Let

$$\boldsymbol{d} = \begin{bmatrix} \bar{\boldsymbol{c}}_{12} - \boldsymbol{c}_{12} \\ (\bar{c}_{22} - c_{22})/2 \\ \bar{\boldsymbol{c}}_{32} - \boldsymbol{c}_{32} \end{bmatrix}.$$

Let $\boldsymbol{e}_k$ be the kth column of the identity. Then $\overline{\mathbf{C}} = \mathbf{C} + \boldsymbol{d}\boldsymbol{e}_k^{\mathsf{T}} + \boldsymbol{e}_k\boldsymbol{d}^{\mathsf{T}}$. This can be put into the form $\overline{\mathbf{C}} = \mathbf{C} + \boldsymbol{w}_1\boldsymbol{w}_1^{\mathsf{T}} - \boldsymbol{w}_2\boldsymbol{w}_2^{\mathsf{T}}$ using the following relationship (note that $\boldsymbol{e}$, below, is an arbitrary column vector, not necessarily $\boldsymbol{e}_k$). Given $\boldsymbol{d}$ and $\boldsymbol{e} \in \mathbb{R}^n$, define

$$\boldsymbol{p} = \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|} + \frac{\boldsymbol{e}}{\|\boldsymbol{e}\|} \quad \text{and} \quad \boldsymbol{q} = \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|} - \frac{\boldsymbol{e}}{\|\boldsymbol{e}\|}.$$

Then we have

$$\boldsymbol{d}\boldsymbol{e}^{\mathsf{T}} + \boldsymbol{e}\boldsymbol{d}^{\mathsf{T}} = \frac{\|\boldsymbol{d}\|\|\boldsymbol{e}\|}{2}\left(\boldsymbol{p}\boldsymbol{p}^{\mathsf{T}} - \boldsymbol{q}\boldsymbol{q}^{\mathsf{T}}\right). \tag{5.1}$$

In our case, $\boldsymbol{e} = \boldsymbol{e}_k$ and $||\boldsymbol{e}|| = 1$. Defining

$$\boldsymbol{w}_1 = \sqrt{\frac{\|\boldsymbol{d}\|}{2}}\left(\frac{\boldsymbol{d}}{\|\boldsymbol{d}\|} + \boldsymbol{e}_k\right) \quad \text{and} \quad \boldsymbol{w}_2 = \sqrt{\frac{\|\boldsymbol{d}\|}{2}}\left(\frac{\boldsymbol{d}}{\|\boldsymbol{d}\|} - \boldsymbol{e}_k\right),$$

it follows from (5.1) that

$$\overline{\mathbf{C}} = \mathbf{C} + \boldsymbol{w}_1\boldsymbol{w}_1^{\mathsf{T}} - \boldsymbol{w}_2\boldsymbol{w}_2^{\mathsf{T}}. \tag{5.2}$$

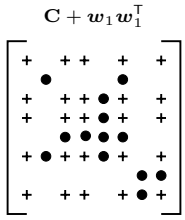


FIG. 5.1. *Modifying* $\mathbf{C}$ *after the first outer product.*

In the dense case, computing (5.2) using a rank-1 update and rank-1 downdate takes $4n^2 + 11n$ flops, independent of k (including the work to compute $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$). If we use Algorithms 1and 2 to make an arbitrary change in the kth row and column of $\mathbf{C}$, the total work is $4n^2 + 8n + 3k^2 - (6nk + 7k)$, which is roughly $4n^2$ when $k = 1$, $n^2 + n$ when $k = n$, and $(7/4)n^2$ when $k = n/2$. Using the rank-2 update/downdate method to modify row and column k of $\mathbf{C}$ can thus take up to four times the work compared to the row addition/deletion presented here.

If the modification requires only the addition or deletion of row and column k, then only Algorithm 1 or 2 needs to be used, but the entire rank-2 update/downdate method presented in this section is still required (about $4n^2$ work, independent of k). Considering only the quadratic terms, Algorithm 1 performs $2n^2 + k^2 - 2nk$ operations, and Algorithm 2 performs $2(n-k)^2$ operations. The row modification methods require much less work, particularly for the row deletion case when $k \approx n$, in which the work drops from $4n^2$ for the rank-2 update/downdate method to nearly no work at all.

In the sparse case, the differences in work and memory usage can be extreme. Both the rank-1 update and downdate could affect the entire matrix $\mathbf{L}$ and could cause catastrophic intermediate fill-in. Consider the row addition case when $k \approx n$ and the new row and column of $\mathbf{C}$ is dense. The factorization of $\overline{\mathbf{C}}$ will still be fairly sparse, since the large submatrix $\mathbf{C}_{11}$ does not change, and remains very sparse. The factor $\mathbf{L}_{11}$ can have as few as $O(n)$ entries. However, after one rank-1 update, the (11)-block of $\mathbf{C} + \boldsymbol{w}_1 \boldsymbol{w}_1^{\mathsf{T}}$ is completely dense, since $\boldsymbol{w}_1$ is a dense column vector. After the rank-1 downdate (with $-\boldsymbol{w}_2 \boldsymbol{w}_2^{\mathsf{T}}$), massive cancellation occurs, and the factorization of $\mathbf{C}_{11}$ is restored to its original sparse nonzero pattern. But the damage has been done, since we require $O(n^2)$ memory to hold the intermediate factorization. This is infeasible in a sparse matrix algorithm. The memory problem could be solved if a single-pass rank-2 update/downdate algorithm were used, but even then, the total work required would be $O(n^2)$, which is much more than the $O(|\overline{\mathbf{L}}|)$ time required for Algorithm 1 in this case. The same problem occurs if the downdate with $-\boldsymbol{w}_2 \boldsymbol{w}_2^{\mathsf{T}}$ is applied first.

Figure 5.1 illustrates the change in $\mathbf{C}$ after the first rank-1 update, if the row modification of Figure 1.2 is performed as the rank-2 modification $\overline{\mathbf{C}} = \mathbf{C} + \boldsymbol{w}_1 \boldsymbol{w}_1^{\mathsf{T}} - \boldsymbol{w}_2 \boldsymbol{w}_2^{\mathsf{T}}$. The vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ have the same nonzero pattern as the change in row k of $\mathbf{C}$. The graph of the matrix $\boldsymbol{w}_1 \boldsymbol{w}_1^{\mathsf{T}}$ is a single clique; its entries are shown as pluses in Figure 5.1. If column 8 of this matrix were modified to become completely dense, then Figure 5.1 would be a full matrix of pluses.

6. Modifying a lower triangular system. We now consider a related operation that can be performed efficiently at the same time that we modify the Cholesky

factorization. Suppose we have a linear system $\mathbf{C}\boldsymbol{x} = \boldsymbol{b}$ and the system is modified, either by changing a row and column of $\mathbf{C}$ as discussed above, or due to a low-rank change $\overline{\mathbf{C}} = \mathbf{C} \pm \mathbf{W}\mathbf{W}^{\mathsf{T}}$ as discussed in [3, 4]. After the factorization is modified, the new factorization $\overline{\mathbf{L}}\overline{\mathbf{D}}\overline{\mathbf{L}}^{\mathsf{T}} = \overline{\mathbf{C}}$ will normally be used to solve a modified linear system $\overline{\mathbf{C}}\overline{\boldsymbol{x}} = \overline{\boldsymbol{b}}$. The complete solution $\overline{\boldsymbol{x}}$ will likely be different in every component because of the backsolve, but a significant reduction of work can be obtained in the forward solve of the lower triangular system $\overline{\mathbf{L}}\overline{\boldsymbol{x}} = \overline{\boldsymbol{b}}$. We thus focus only on this lower triangular system, not the complete system.

First, consider the simpler case of a low-rank change of a dense matrix. As shown below, it takes double the work to modify the solution instead of computing it from scratch, but the dense matrix method can be used for submatrices in the sparse case, resulting in a significant reduction in work. Suppose we have the system $\mathbf{L}\boldsymbol{x} = \boldsymbol{b}$, including its solution $\boldsymbol{x}$, and the new linear system is $\overline{\mathbf{L}}\overline{\boldsymbol{x}} = \overline{\boldsymbol{b}}$. Combining these two equations gives

$$\overline{\mathbf{L}}\overline{\boldsymbol{x}} = \overline{\boldsymbol{b}} - \boldsymbol{b} + \mathbf{L}\boldsymbol{x} = \Delta\boldsymbol{b} + \mathbf{L}\boldsymbol{x},$$

where $\Delta\boldsymbol{b} = \overline{\boldsymbol{b}} - \boldsymbol{b}$. Suppose we are given $\mathbf{L}$, $\boldsymbol{x}$, and $\Delta\boldsymbol{b}$. The matrix $\overline{\mathbf{L}}$ is computed column-by-column by either the low-rank update/downdate algorithm in [3, 4] or the row and column modification algorithm discussed in this paper. We can combine the modification of $\mathbf{L}$ with the computation of $\overline{\boldsymbol{x}}$, as shown in Algorithm 3.

ALGORITHM 3 (DENSE MODIFICATION OF $\mathbf{L}\boldsymbol{x} = \boldsymbol{b}$ TO SOLVE $\overline{\mathbf{L}}\overline{\boldsymbol{x}} = \overline{\boldsymbol{b}}$).

$\overline{\boldsymbol{x}} = \Delta\boldsymbol{b}$
for $j = 1$ to n **do**
  $\overline{\boldsymbol{x}} = \overline{\boldsymbol{x}} + (\mathbf{L})_{*j} x_j$
  compute the new column $(\overline{\mathbf{L}})_{*j}$
  $(\overline{\boldsymbol{x}})_{j+1\ldots n} = (\overline{\boldsymbol{x}})_{j+1\ldots n} - (\overline{\mathbf{L}})_{j+1\ldots n,j}\overline{x}_j$
end for
end Algorithm 3

The total work to modify the solution to $\overline{\mathbf{L}}\overline{\boldsymbol{x}} = \overline{\boldsymbol{b}}$ is roughly $2n^2$, as compared to n^2 work to solve $\overline{\mathbf{L}}\overline{\boldsymbol{x}} = \overline{\boldsymbol{b}}$ from scratch. One would never use this method if $\mathbf{L}$ and $\boldsymbol{b}$ are dense.

Now consider the sparse case. Suppose we have a low-rank sparse update of $\mathbf{L}$. The columns that change in $\overline{\mathbf{L}}$ correspond to a single path $\overline{\mathcal{P}}$ in the elimination tree for a rank-1 update. Every entry in these specific columns is modified. For a rank-r update, where $r > 1$, the columns that change correspond to a set of paths in the elimination tree, which we also will refer to as $\overline{\mathcal{P}}$ in this more general case. In both cases, the nonzero pattern of each column j in the path $\overline{\mathcal{P}}$ is a subset of the path [20]. We can thus partition the matrices $\mathbf{L}$ and $\overline{\mathbf{L}}$ into two parts, according to the set $\overline{\mathcal{P}}$. The original system is

$$\begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix},$$

where the matrix $\mathbf{L}_{22}$ consists of all the rows and columns corresponding to nodes in the path $\overline{\mathcal{P}}$. If the changes in the right-hand side $\overline{\boldsymbol{b}}$ are also constrained to the set $\overline{\mathcal{P}}$, the new linear system is

$$\begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{12} & \overline{\mathbf{L}}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \overline{\boldsymbol{x}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \overline{\boldsymbol{b}}_2 \end{bmatrix}.$$

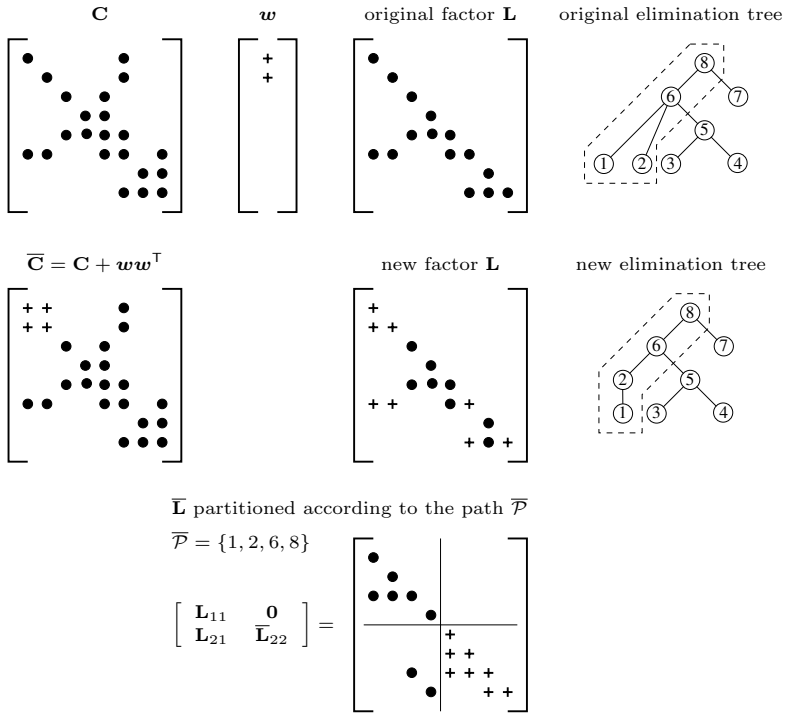


FIG. 6.1. *Modifying* $\mathbf{L}\boldsymbol{x} = \boldsymbol{b}$ *after a rank-*1 *update.*

Note that $\mathbf{L}_{11}$, $\mathbf{L}_{12}$, and $\boldsymbol{b}_1$ do not change, and thus $\boldsymbol{x}_1$ does not change. If the change in the right-hand side is arbitrary (not constrained to $\overline{\mathcal{P}}$, for example), then $\boldsymbol{x}_1$ changes and no work is saved over solving $\overline{\mathbf{L}}\overline{\boldsymbol{x}} = \overline{\boldsymbol{b}}$ from scratch.

We have made a sparse change to both the matrix $\mathbf{L}$ and the right-hand side. The solution to the subsystem $\mathbf{L}_{11}\boldsymbol{x}_1 = \boldsymbol{b}_1$ does not change. We have

$$\begin{aligned} \mathbf{L}_{21}\boldsymbol{x}_1 + \mathbf{L}_{22}\boldsymbol{x}_2 &= \boldsymbol{b}_2, \\ \mathbf{L}_{21}\boldsymbol{x}_1 + \overline{\mathbf{L}}_{22}\overline{\boldsymbol{x}}_2 &= \overline{\boldsymbol{b}}_2. \end{aligned}$$

We can apply Algorithm 3 to the subsystem

$$\overline{\mathbf{L}}_{22}\overline{\boldsymbol{x}}_2 = \Delta\boldsymbol{b}_2 + \mathbf{L}_{22}\boldsymbol{x}_{22},$$

where $\Delta\boldsymbol{b}_2 = \overline{\boldsymbol{b}}_2 - \boldsymbol{b}_2$, to obtain the new solution $\overline{\boldsymbol{x}}_2$. Algorithm 3 takes $4|\overline{\mathbf{L}}_{22}| + O(\overline{\mathcal{P}})$ flops to compute $\overline{\boldsymbol{x}}_2$. An additional $4r|\overline{\mathbf{L}}_{22}| + O(\overline{\mathcal{P}})$ flops are required to update $\overline{\mathbf{L}}_{22}$. Solving $\overline{\mathbf{L}}\overline{\boldsymbol{x}} = \overline{\boldsymbol{b}}$ after $\overline{\mathbf{L}}$ is updated takes $2|\overline{\mathbf{L}}|$ flops. If the path $\overline{\mathcal{P}}$ is short, making a sparse modification to the old solution $\boldsymbol{x}$ to obtain the new solution $\overline{\boldsymbol{x}}$ during the update of $\overline{\mathbf{L}}$ takes much less work than solving the new linear system after $\overline{\mathbf{L}}$ is updated.

Figure 6.1 shows a matrix $\mathbf{C}$, its factorization, and its elimination tree after a rank-1 update, using just the first column of $\mathbf{W}$ from the example shown in Figure 1.1. As in our other figures, a plus denotes an entry that changes, or an entry of $\boldsymbol{w}$. The subtree of the original elimination tree consisting of nodes $\{1, 2, 6, 8\}$ becomes a single path $\overline{\mathcal{P}} = \{1, 2, 6, 8\}$ in the new elimination tree. The rows and columns of $\overline{\mathbf{L}}$ can be

partitioned according to the path $\overline{\mathcal{P}}$, as shown in Figure 6.1. We do not perform this permutation, of course, but only illustrate it here. The algorithm that updates $\mathbf{L}$ and $\boldsymbol{x}$ accesses only columns $\{1, 2, 6, 8\}$ of $\mathbf{L}$ (the submatrix $\mathbf{L}_{22}$) and the same rows of $\boldsymbol{x}$ and $\boldsymbol{b}$.

Finally, consider the case discussed in this paper of modifying row and column k of $\mathbf{C}$. If we add row k, the original lower triangular system is

$$\begin{bmatrix} \mathbf{L}_{11} & & \\ \mathbf{0}^{\mathsf{T}} & 1 & \\ \mathbf{L}_{31} & \mathbf{0} & \mathbf{L}_{33} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ x_2 \\ \boldsymbol{x}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ b_2 \\ \boldsymbol{b}_3 \end{bmatrix}. \tag{6.1}$$

The new lower triangular system is

$$\begin{bmatrix} \mathbf{L}_{11} & & \\ \bar{\boldsymbol{l}}_{12}^{\mathsf{T}} & 1 & \\ \mathbf{L}_{31} & \bar{\boldsymbol{l}}_{32} & \overline{\mathbf{L}}_{33} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \overline{x}_2 \\ \overline{\boldsymbol{x}}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \bar{b}_2 \\ \overline{\boldsymbol{b}}_3 \end{bmatrix}, \tag{6.2}$$

where we assume $\boldsymbol{b}_1$ does not change, and the entries that change in $\boldsymbol{b}_3$ are a subset of the path $\overline{\mathcal{P}}$. Let $\Delta\boldsymbol{b}_3 = \overline{\boldsymbol{b}}_3 - \boldsymbol{b}_3$. The term $\overline{x}_2$ can be computed as

$$\overline{x}_2 = \bar{b}_2 - \bar{\boldsymbol{l}}_{12}^{\mathsf{T}} \boldsymbol{x}_1.$$

The (33)-blocks of (6.1) and (6.2) give the equations

$$\begin{aligned} \mathbf{L}_{33}\boldsymbol{x}_3 &= \boldsymbol{b}_3 - \mathbf{L}_{31}\boldsymbol{x}_1, \\ \overline{\mathbf{L}}_{33}\overline{\boldsymbol{x}}_3 &= (\boldsymbol{b}_3 - \mathbf{L}_{31}\boldsymbol{x}_1) + (\Delta\boldsymbol{b}_3 - \bar{\boldsymbol{l}}_{32}\overline{x}_2). \end{aligned} \tag{6.3}$$

The change in the right-hand side of this system is $\Delta\boldsymbol{b}_3 - \bar{\boldsymbol{l}}_{32}\overline{\boldsymbol{x}}_2$. Since the nonzero pattern of $\bar{\boldsymbol{l}}_{32}$ is a subset of $\overline{\mathcal{P}}$, this computation fits the same requirements for the sparse change in $\boldsymbol{x}$ due to a sparse low-rank change in $\mathbf{L}$ and $\boldsymbol{b}$, discussed above. The row deletion case is analogous.

The number of flops in Algorithm 3 to modify the solution to $\overline{\mathbf{L}}\overline{\boldsymbol{x}} = \overline{\boldsymbol{b}}$ is roughly the same as a rank-1 update to compute the modified $\overline{\mathbf{L}}$. However, the run time is not doubled compared to a stand-alone rank-1 update. At step j, for each j in the path $\overline{\mathcal{P}}$, the rank-1 update must read column j of $\mathbf{L}$, modify it, and then write the jth column of $\overline{\mathbf{L}}$ back into memory. No extra memory traffic to access $\overline{\mathbf{L}}$ is required to compute the solution to the lower triangular system using (6.3). With current technology, memory traffic can consume more time than flops. Thus, when modifying the kth row and column of $\mathbf{C}$, or performing a rank-r update/downdate to $\mathbf{C}$, we can update the solution to the lower triangular system at almost no extra cost. In our target application [5], solving the linear system $\mathbf{C}\boldsymbol{x} = \boldsymbol{b}$, given its $\mathbf{LDL}^{\mathsf{T}}$ factorization, can often be the dominant step. The method presented here can cut this time almost in half since the forward solve time is virtually eliminated, leaving us with the time for the upper triangular backsolve.

7. Experimental results. To illustrate the performance of the algorithms developed in this paper in a specific application, Table 7.1 gives the flops associated with the solution of the four largest problems in the Netlib linear programming test set using the LPDASA [5]. The problems passed to the solver were first simplified using the ILOG CPLEX [2] version 7.0 presolve routine. An LP presolver [1] preprocesses the problem by removing redundant constraints and variables whose values can be

TABLE 7.1
Experimental results.

Problem	Performance	Forward solve	Column updates	Row deletions
DFL001	number:		2629	87
n: 3881	avg. mod. rank		1.2	1
	avg. mod. flops		2.43×10^6	1.64×10^6
	avg. solve flops	1.46×10^6	2.03×10^6	1.64×10^6
PDS20	number:		1736	65
n: 10214	avg. mod. rank		3.5	1
	avg. mod. flops		3.37×10^6	2.13×10^6
	avg. solve flops	1.11×10^6	1.21×10^6	2.12×10^6
KEN18	number:		2799	23
n: 39856	avg. mod. rank		15.6	1
	avg. mod. flops		61.7×10^3	2242
	avg. solve flops	195.8×10^3	7155	2075
OSA60	number:		71	277
n: 10209	avg. mod. rank		58.9	1
	avg. mod. flops		4415	62
	avg. solve flops	11.0×10^3	270	37

easily determined. Hence, the number of rows n of the problems listed in the first column of Table 7.1 is much smaller than the number of rows in the original Netlib problems.

The third column (labeled "Forward solve") gives the average number of flops that would have been required if forward solves were done in a conventional way, by a forward elimination process. This number is simply twice the average number of nonzeros in $\mathbf{L}$. As the LP is solved, the sparsity of $\mathbf{L}$ changes slightly due to changes in the current basis. Hence, we give the average flops needed for a conventional forward solve, which can be compared with the average flops in modifying the solution using the technique developed in this paper. The problems are sorted according to this column.

The fourth column of the table, entitled "Column updates," lists the number of rank-r column updates performed (of the form $\mathbf{C} + \mathbf{W}\mathbf{W}^{\mathsf{T}}$), the average rank r of those updates, the flops required to modify $\mathbf{L}$, and the flops required to modify the solution to the forward solve when $\mathbf{L}$ changes as a result of a column update. Since we have developed [4] a multiple-rank approach for performing column updates, the average ranks listed are all greater than 1. They are near 1 for the densest problem DFL001 and near 60 for the sparsest problem OSA60. Recall that the column update requires about $4r$ flops per entry in $\mathbf{L}$ that change. Modifying the forward solve takes 4 flops per entry in $\mathbf{L}$ that change. The average flops associated with the modification of $\mathbf{L}$ is thus always greater than the flops associated with the forward solve update, especially for multiple rank column updates. The conventional forward solve requires 2 flops for each nonzero entry in $\mathbf{L}$, whether they change or not. In the worst case, when all of the entries in $\mathbf{L}$ change, modifying the forward solve takes no more than twice the work of the conventional forward solve, and a column update takes no more than $2r$ times the work of a conventional forward solve.

The last column of the table, entitled "Row deletions," lists the number of deletions of a row (and column) from $\mathbf{C}$, the average number of flops required to modify $\mathbf{L}$ after deleting a row from $\mathbf{C}$, and the average number of flops required to modify the solution to the forward solve when $\mathbf{L}$ changes as a result of a row deletion. Since

the row deletion algorithm developed in this paper is a single-rank process, the ranks listed are all 1.

When the matrix $\mathbf{L}$ is very sparse (such as OSA60), both the column update and row deletion methods modify only a small portion of the matrix. The elimination tree tends to be short and wide, with short paths from any node to the root. Thus, for very sparse problems the conventional forward solve (which accesses all of $\mathbf{L}$) takes much more work than either the column update or the row deletion. For the OSA60 matrix, the conventional forward solve takes about 40 times the work compared to modifying the forward solve during a column update.

For a matrix $\mathbf{L}$ that is fairly dense (such as DFL001), the elimination tree tends to be tall and thin, and the column updates or row deletions modify most entries in $\mathbf{L}$. In this case, the work required to modify the forward solve is more on average than that of a full forward solve, but it is never more than twice the work. A combined update of the factorization and forward solve cuts the memory traffic at least in half compared with an update of $\mathbf{L}$ followed by a conventional forward solve. The update accesses only the parts of $\mathbf{L}$ that change, whereas the conventional forward solve accesses all of $\mathbf{L}$. The performance of most modern computers is substantially affected by the amount of memory transfers between cache and main memory, so cutting memory traffic at least in half at the cost of at most doubling the flop count will normally lead to an overall improvement in performance, even when the matrix is fairly dense.

8. Summary. We have presented a method for modifying the sparse Cholesky factorization of a symmetric positive definite matrix $\mathbf{C}$ after a row and column of $\mathbf{C}$ have been modified. One algorithm, the sparse row addition, is optimal. The corresponding row deletion algorithm is not optimal but takes no more time than the row addition. Although changing a row and column of $\mathbf{C}$ can be cast as rank-2 change of the form $\mathbf{C} + \boldsymbol{w}_1\boldsymbol{w}_1^{\mathsf{T}} - \boldsymbol{w}_2\boldsymbol{w}_2^{\mathsf{T}}$, the latter is impractical in a sparse context. We have shown how to modify the solution to a lower triangular system $\mathbf{L}\boldsymbol{x} = \boldsymbol{b}$ when the matrix $\mathbf{L}$ changes as a result of either the row addition/deletion operation discussed here or an update/downdate of the form $\mathbf{C} \pm \mathbf{W}\mathbf{W}^{\mathsf{T}}$ described in our previous papers [3, 4]. By postponing the symbolic downdate, the memory usage has been reduced by 25% (assuming 8-byte floating-point values and 4-byte integers), compared with the column update/downdate methods described in [3, 4], which also store the multiplicities. Together, the row addition/deletion algorithms and the column update/downdate algorithms form a useful suite of tools for modifying a sparse Cholesky factorization and for solving a sparse system of linear equations. Using these algorithms, the linear programming solver LPDASA is able to achieve an overall performance that rivals, and sometimes exceeds, the performance of current commercially available solvers [5].

REFERENCES

[1] E. D. Andersen and K. D. Andersen, *Presolving in linear programming*, Math. Program., 71 (1995), pp. 221–245.

[2] R. E. Bixby, *Progress in linear programming*, ORSA J. Comput., 6 (1994), pp. 15–22.

[3] T. A. Davis and W. W. Hager, *Modifying a sparse Cholesky factorization*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 606–627.

[4] T. A. Davis and W. W. Hager, *Multiple-rank modifications of a sparse Cholesky factorization*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 997–1013.

[5] T. A. Davis and W. W. Hager, *A sparse proximal implementation of the LP dual active set algorithm*, Math. Program., submitted; also available online from http://www.math.ufl.edu/∼hager.

[6] S. C. Eisenstat, M. C. Gursky, M. H. Schultz, and A. H. Sherman, *Yale sparse matrix package* I: *The symmetric codes*, Internat. J. Numer. Methods Engrg., 18 (1982), pp. 1145–1151.

[7] A. George and J. W. H. Liu, *An optimal algorithm for symbolic factorization of symmetric matrices*, SIAM J. Comput., 9 (1980), pp. 583–593.

[8] A. George and J. W. H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1981.

[9] J. R. Gilbert, *Predicting structure in sparse matrix computations*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 62–79.

[10] J. R. Gilbert and T. Peierls, *Sparse partial pivoting in time proportional to arithmetic operations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 862–874.

[11] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, *Methods for modifying matrix factorizations*, Math. Comp., 28 (1974), pp. 505–535.

[12] W. W. Hager, *Updating the inverse of a matrix*, SIAM Rev., 31 (1989), pp. 221–239.

[13] W. W. Hager, *The dual active set algorithm*, in Advances in Optimization and Parallel Computing, P. M. Pardalos, ed., North–Holland, Amsterdam, 1992, pp. 137–142.

[14] W. W. Hager, *The LP dual active set algorithm*, in High Performance Algorithms and Software in Nonlinear Optimization, R. D. Leone, A. Murli, P. M. Pardalos, and G. Toraldo, eds., Kluwer Academic Publishers, Norwell, MA, 1998, pp. 243–254.

[15] W. W. Hager, *The dual active set algorithm and its application to linear programming*, Comput. Optim. Appl., 21 (2002), pp. 263–275.

[16] W. W. Hager, *The dual active set algorithm and the iterative solution of linear programs*, in Novel Approaches to Hard Discrete Optimization, P. M. Pardalos and H. Wolkowicz, eds., Kluwer Academic Publishers, Norwell, MA, 2003, pp. 95–107.

[17] W. W. Hager and D. W. Hearn, *Application of the dual active set algorithm to quadratic network optimization*, Comput. Optim. Appl., 1 (1993), pp. 349–373.

[18] J. W. H. Liu, *A compact row storage scheme for Cholesky factors using elimination trees*, ACM Trans. Math. Software, 12 (1986), pp. 127–148.

[19] J. W. H. Liu, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 134–172.

[20] R. Schreiber, *A new implementation of sparse Gaussian elimination*, ACM Trans. Math. Software, 8 (1982), pp. 256–276.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 3, pp. 640–659

SOAR: A SECOND-ORDER ARNOLDI METHOD FOR THE SOLUTION OF THE QUADRATIC EIGENVALUE PROBLEM*

ZHAOJUN BAI† AND YANGFENG SU‡

Abstract. We first introduce a *second-order Krylov subspace* $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$ based on a pair of square matrices $\mathbf{A}$ and $\mathbf{B}$ and a vector $\mathbf{u}$. The subspace is spanned by a sequence of vectors defined via a second-order linear homogeneous recurrence relation with coefficient matrices $\mathbf{A}$ and $\mathbf{B}$ and an initial vector $\mathbf{u}$. It generalizes the well-known Krylov subspace $\mathcal{K}_n(\mathbf{A}; \mathbf{v})$, which is spanned by a sequence of vectors defined via a first-order linear homogeneous recurrence relation with a single coefficient matrix $\mathbf{A}$ and an initial vector $\mathbf{v}$. Then we present a second-order Arnoldi (SOAR) procedure for generating an orthonormal basis of $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$. By applying the standard Rayleigh–Ritz orthogonal projection technique, we derive an SOAR method for solving a large-scale quadratic eigenvalue problem (QEP). This method is applied to the QEP directly. Hence it preserves essential structures and properties of the QEP. Numerical examples demonstrate that the SOAR method outperforms convergence behaviors of the Krylov subspace–based Arnoldi method applied to the linearized QEP.

Key words. quadratic eigenvalue problem, second-order Krylov subspace, second-order Arnoldi procedure, Rayleigh–Ritz orthogonal projection

AMS subject classifications. 65F15, 65F30

DOI. 10.1137/S0895479803438523

1. Introduction.

The Krylov subspace

$$\mathcal{K}_n(\mathbf{A}; \mathbf{v}) = \text{span}\{\mathbf{v}, \mathbf{A}\mathbf{v}, \mathbf{A}^2\mathbf{v}, \ldots, \mathbf{A}^{n-1}\mathbf{v}\} \tag{1.1}$$

based on a square matrix $\mathbf{A}$ and a vector $\mathbf{v}$ plays an indispensable role in modern numerical techniques for solving large-scale matrix computation problems, such as the linear eigenvalue problem of the form $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$. A Krylov subspace–based method is often the method of choice due to its simplicity, its availability of reliable and efficient processes for generating its orthonormal basis, and the superiority of convergence behaviors [5, 6, 12, 15, 16]. Many state-of-the-art Krylov subspace methods for solving large-scale eigenvalue problems are presented in [3].

The generalized eigenvalue problem of the form $\mathbf{A}\mathbf{x} = \lambda\mathbf{B}\mathbf{x}$ must be reduced, explicitly or implicitly, to the linear eigenvalue problem in a form such as $(\mathbf{B}^{-1}\mathbf{A})\mathbf{x} = \lambda\mathbf{x}$, and then a Krylov subspace–based method can be applied. The quadratic eigenvalue problem (QEP) of the form

$$(\lambda^2\mathbf{M} + \lambda\mathbf{D} + \mathbf{K})\mathbf{x} = \mathbf{0} \tag{1.2}$$

is usually processed in two stages, as recommended in most literature, public domain packages, and proprietary software today. At the first stage, it transforms the QEP into an equivalent generalized eigenvalue problem:

$$\mathbf{C}\mathbf{y} = \lambda\mathbf{G}\mathbf{y}, \tag{1.3}$$

*Received by the editors December 9, 2003; accepted for publication (in revised form) by H. A. van der Vorst July 2, 2004; published electronically March 3, 2005.
http://www.siam.org/journals/simax/26-3/43852.html

†Department of Computer Science and Department of Mathematics, University of California, Davis, CA 95616 (bai@cs.ucdavis.edu). The research of this author was supported in part by the National Science Foundation under grant 0220104.

‡Department of Mathematics, Fudan University, Shanghai 200433, China (yfsu@fudan.edu.cn). The research of this author was supported in part by NSFC research project 10001009 and NSFC research key project 90307017.

where $\mathbf{y}^{\mathrm{T}} = \begin{bmatrix} \lambda\mathbf{x}^{\mathrm{T}} & \mathbf{x}^{\mathrm{T}} \end{bmatrix}$, and $\mathbf{C}$ and $\mathbf{G}$ are in forms such as

$$\mathbf{C} = \begin{bmatrix} -\mathbf{D} & -\mathbf{K} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix},$$

where we assume throughout the report that $\mathbf{M}$ is nonsingular. At the second stage, it reduces the generalized eigenvalue problem (1.3) to a linear eigenvalue problem "$\mathbf{Ax} = \lambda\mathbf{x}$" and then applies a Krylov subspace–based method. Such an approach takes advantages of Krylov subspace–based methods, such as the fast convergence rate and the simultaneous convergence of a group of eigenvalues. However, it also suffers some disadvantages, such as having to solve the generalized eigenvalue problem (1.3) of twice the dimension of the original QEP and, more importantly, the loss of the original structures of the QEP in the process of linearization. For example, when coefficient matrices $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ are symmetric positive definite, the transformed generalized eigenvalue problem (1.3) has to be either intrinsically nonsymmetric, where one of $\mathbf{C}$ and $\mathbf{G}$ has to be nonsymmetric, or symmetric indefinite, where both $\mathbf{C}$ and $\mathbf{G}$ are symmetric but neither will be positive definite. Subsequently, essential spectral properties of the QEP are not guaranteed to be preserved. The reader is referred to [24] for a recent survey on theory, applications, and algorithms of the QEP.

For years, researchers have been studying numerical methods which can be applied to the large-scale QEP directly. In these methods, they do not transform the QEP into an equivalent linear form; instead, they project the QEP onto a properly chosen low-dimensional subspace to reduce to a QEP directly with matrix dimension of lower order. The reduced QEP problem can then be solved by a standard dense matrix technique. The Jacobi–Davidson method [17, 18] is one such method. The method targets one eigenvalue at a time with local convergence versus Krylov subspace methods in which a group of eigenvalues is approximated with global convergence. A direct Krylov-type subspace method with a generalized Arnoldi procedure is briefly described in [13]. However, the procedure presented in [13] in fact does not compute an orthonormal basis of the desired Krylov-type subspace. In [7], Arnoldi- and Lanczos-type processes are developed to construct projections of the QEP. The convergence of these methods is usually slower than a Krylov subspace method applied to the mathematically equivalent linear eigenvalue problem. Finally, a subspace approximation method that uses perturbation theory of the QEP was recently presented in [8]. The success of the method is strongly dependent on the initial approximation, although Rayleigh quotient iteration can be used for acceleration.

Motivated by striking an ideal method which not only can be applied to the QEP directly to preserve the essential structures of the QEP but also achieves the superior global convergence behaviors of Krylov subspace methods via linearization, in this paper, we first introduce a second-order Krylov subspace $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$ based on a pair of square matrices $\mathbf{A}$ and $\mathbf{B}$ and a vector $\mathbf{u}$. The basis vectors of the subspace are defined via a linear homogeneous recurrence of degree 2 with coefficient matrices $\mathbf{A}$ and $\mathbf{B}$. Consequently, a second-order Arnoldi (SOAR) procedure is presented for generating an orthonormal basis of $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$. As an application of the SOAR procedure, a Rayleigh–Ritz orthogonal projection technique based on $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$ is discussed for finding a few of the largest magnitude eigenvalues and the corresponding eigenvectors of the large-scale QEP (1.2). This method is applied to the QEP directly. Hence it preserves essential structures and properties of the QEP. Numerical examples presented in section 5 demonstrate that the new QEP solver outperforms

convergence behaviors of the Krylov subspace–based Arnoldi method when applied to the linearized QEP.

In order to solve the large-scale QEP and, more generally, the matrix polynomial eigenvalue problem efficiently, the necessity for the extension of the standard Krylov subspace to explicitly involve more than one matrix has been recognized. In section 2, we will see that the definition of the subspace $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$ is a natural extension in the context of solving the QEP by a projection technique. It has been an interesting problem to find a scheme which can efficiently construct an orthonormal basis of $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$ that is comparable to the Arnoldi process for generating an orthonormal basis of the standard Krylov subspace $\mathcal{K}_n(\mathbf{A}; \mathbf{u})$. The first procedure presented in this paper is inspired by the work of Su and Craig [22], to which we are gratefully indebted.

The rest of this report is organized as follows. In section 2, we introduce the second-order Krylov subspace $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$ and a simple SOAR procedure for generating an orthonormal basis of the subspace. In section 3, we discuss the possible deflation and breakdown situations of the SOAR procedure, and we present a revised version of the SOAR procedure with deflation and memory saving. In section 4, we present a Rayleigh–Ritz procedure for solving the QEP (1.2). For completeness, we also present the basic Arnoldi method for solving the equivalent generalized eigenvalue problem (1.3). Numerical examples are presented in section 5. Discussion and future work are in section 6.

Throughout the paper, we follow the notational convention commonly used in matrix computation literature. Specifically, we use boldface letters to denote vectors (lower cases) and matrices (upper cases), $\mathbf{I}$ for the identity matrix, $\mathbf{e}_j$ for the jth column of the identity matrix $\mathbf{I}$, and $\mathbf{0}$ for zero vectors and matrices. The dimensions of these vectors and matrices are conformed with dimensions used in the context. We use $\cdot^{\mathrm{T}}$ to denote the transpose. N denotes the order of the original matrix triplet $(\mathbf{M}, \mathbf{D}, \mathbf{K})$ and associated QEP (1.2). $\mathrm{span}\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n\}$ and $\mathrm{span}\{\mathbf{Q}\}$ denote the space spanned by the vector sequence $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$ and the columns of the matrix $\mathbf{Q}$, respectively. $\|\cdot\|_1$ and $\|\cdot\|_2$ denote the 1-norm and 2-norm, respectively, for vector or matrix. $\mathbf{x}(i:j)$, as used in MATLAB, denotes the ith to jth entries of the vector $\mathbf{x}$. $\mathbf{A}(i:j, k:\ell)$ denotes the submatrix of $\mathbf{A}$ by the intersection of rows i to j and columns k to ℓ.

2. A second-order Krylov subspace. In this section, we first define a generalized Krylov subspace induced by a pair of matrices $\mathbf{A}$ and $\mathbf{B}$ and a vector $\mathbf{u}$. Then we discuss the motivation for such a generalization and present an Arnoldi-like procedure for generating an orthonormal basis of the generalized Krylov subspace.

DEFINITION 2.1. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be square matrices of order* N*, and let* $\mathbf{u} \neq \mathbf{0}$ *be an* N *vector. Then the sequence*

$$\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{n-1}, \tag{2.1}$$

where

$$\begin{aligned} \mathbf{r}_0 &= \mathbf{u}, \\ \mathbf{r}_1 &= \mathbf{A}\mathbf{r}_0, \\ \mathbf{r}_j &= \mathbf{A}\mathbf{r}_{j-1} + \mathbf{B}\mathbf{r}_{j-2} \quad \textit{for } j \geq 2, \end{aligned}$$

is called a second-order Krylov sequence *based on* $\mathbf{A}$*,* $\mathbf{B}$*, and* $\mathbf{u}$*. The space*

$$\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u}) = \mathrm{span}\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{n-1}\}$$

is called an nth second-order Krylov subspace.

First, we note that the subspace $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$ generalizes the standard Krylov subspace $\mathcal{K}_n(\mathbf{A}; \mathbf{u})$ in the way that when $\mathbf{B}$ is a zero matrix, the second-order Krylov subspace is the standard Krylov subspace, namely,

$$\mathcal{G}_n(\mathbf{A}, \mathbf{0}; \mathbf{u}) = \mathcal{K}_n(\mathbf{A}; \mathbf{u}).$$

Second, we know that the Krylov subspace $\mathcal{K}_n(\mathbf{A}; \mathbf{u})$ has an important characterization in terms of matrix polynomials, which forms a foundation for convergence analysis of a Krylov subspace–based method. There is a similar one for the second-order Krylov subspace $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$. With the starting vector $\mathbf{u}$, the first few vectors in the second-order Krylov sequence can be written as

$$\begin{aligned}
&\mathbf{r}_0 = \mathbf{u},\\
&\mathbf{r}_1 = \mathbf{A}\mathbf{u},\\
&\mathbf{r}_2 = (\mathbf{A}^2 + \mathbf{B})\mathbf{u},\\
&\mathbf{r}_3 = (\mathbf{A}^3 + \mathbf{A}\mathbf{B} + \mathbf{B}\mathbf{A})\mathbf{u},\\
&\mathbf{r}_4 = (\mathbf{A}^4 + \mathbf{A}^2\mathbf{B} + \mathbf{A}\mathbf{B}\mathbf{A} + \mathbf{B}\mathbf{A}^2 + \mathbf{B}^2)\mathbf{u}.
\end{aligned}$$

In general, the jth vector $\mathbf{r}_j$ in the second-order Krylov sequence defined by a linear homogeneous recurrence relation of degree 2 with coefficient matrices $\mathbf{A}$ and $\mathbf{B}$ can be written as

$$\mathbf{r}_j = p_j(\mathbf{A}, \mathbf{B})\mathbf{u},$$

where $p_j(\alpha, \beta)$ are polynomials in α and β, defined by the recurrence

$$p_j(\alpha, \beta) = \alpha \cdot p_{j-1}(\alpha, \beta) + \beta \cdot p_{j-2}(\alpha, \beta)$$

with $p_0(\alpha, \beta) \equiv 1$ and $p_1(\alpha, \beta) = \alpha$.

We now discuss the motivation for the definition of the second-order Krylov subspace $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$ in the context of solving the QEP (1.2). Recall that the QEP (1.2) can be transformed into an equivalent generalized eigenvalue problem (1.3). If one applies a Krylov subspace technique to (1.3), then an associated Krylov subspace would naturally be

$$\mathcal{K}_n(\mathbf{H}; \mathbf{v}) = \operatorname{span}\{\mathbf{v}, \mathbf{H}\mathbf{v}, \mathbf{H}^2\mathbf{v}, \ldots, \mathbf{H}^{n-1}\mathbf{v}\}, \tag{2.2}$$

where $\mathbf{v}$ is a starting vector of length $2N$, and

$$\mathbf{H} = \mathbf{G}^{-1}\mathbf{C} = \begin{bmatrix} -\mathbf{M}^{-1}\mathbf{D} & -\mathbf{M}^{-1}\mathbf{K} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}. \tag{2.3}$$

Let $\mathbf{A} = -\mathbf{M}^{-1}\mathbf{D}$, $\mathbf{B} = -\mathbf{M}^{-1}\mathbf{K}$, and $\mathbf{v} = [\mathbf{u}^{\mathrm{T}} \ \ \mathbf{0}]^{\mathrm{T}}$; then we immediately derive that the second-order Krylov vectors $\{\mathbf{r}_j\}$ of length N defined in (2.1) and the standard Krylov vectors $\{\mathbf{H}^j\mathbf{v}\}$ of length $2N$ defined in (2.2) are related as the following form:

$$\begin{bmatrix} \mathbf{r}_j \\ \mathbf{r}_{j-1} \end{bmatrix} = \mathbf{H}^j\mathbf{v} \quad \text{for } j \geq 1. \tag{2.4}$$

In other words, the generalized Krylov sequence $\{\mathbf{r}_j\}$ *defines* the entire standard Krylov sequence based on $\mathbf{H}$ and $\mathbf{v}$. Equation (2.4) indicates that the subspace $\mathcal{G}_j(\mathbf{A}, \mathbf{B}; \mathbf{u})$ of $\mathcal{R}^N$ should be able to provide sufficient information to let us directly

work with the QEP, instead of using the subspace $\mathcal{K}_n(\mathbf{H};\mathbf{v})$ of $\mathcal{R}^{2N}$ for the linearized eigenvalue problem (1.3). We will discuss this further in section 4.

We now turn to the question of how to construct an orthonormal basis $\{\mathbf{q}_i\}$ of $\mathcal{G}_j(\mathbf{A},\mathbf{B};\mathbf{u})$. Namely,

$$\operatorname{span}\{\mathbf{q}_1,\mathbf{q}_2,\ldots,\mathbf{q}_j\} = \mathcal{G}_j(\mathbf{A},\mathbf{B};\mathbf{u}) \quad \text{for } j \geq 1.$$

The following is a procedure to implicitly apply to the sequence of the second-order Krylov vectors $\{\mathbf{r}_j\}$ to generate an orthonormal basis $\{\mathbf{q}_1,\mathbf{q}_2,\ldots,\mathbf{q}_j\}$. Later we will see that it is an Arnoldi-like procedure. We call it an SOAR (second-order Arnoldi) procedure.

ALGORITHM 1. *SOAR procedure.*

1. $\mathbf{q}_1 = \mathbf{u}/\|\mathbf{u}\|_2$
2. $\mathbf{p}_1 = \mathbf{0}$
3. **for** $j = 1,2,\ldots,n$ **do**
4. $\quad \mathbf{r} = \mathbf{A}\mathbf{q}_j + \mathbf{B}\mathbf{p}_j$
5. $\quad \mathbf{s} = \mathbf{q}_j$
6. $\quad$ **for** $i = 1,2,\ldots,j$ **do**
7. $\quad\quad t_{ij} = \mathbf{q}_i^{\mathrm{T}}\mathbf{r}$
8. $\quad\quad \mathbf{r} := \mathbf{r} - \mathbf{q}_i t_{ij}$
9. $\quad\quad \mathbf{s} := \mathbf{s} - \mathbf{p}_i t_{ij}$
10. $\quad$ **end for**
11. $\quad t_{j+1\,j} = \|\mathbf{r}\|_2$
12. $\quad$ **if** $t_{j+1\,j} = 0$, **stop**
13. $\quad \mathbf{q}_{j+1} = \mathbf{r}/t_{j+1\,j}$
14. $\quad \mathbf{p}_{j+1} = \mathbf{s}/t_{j+1\,j}$
15. **end for**

We note that matrices $\mathbf{A}$ and $\mathbf{B}$ are referenced only via the matrix-vector multiplications in line 4 of the algorithm above. Therefore, it is ideal for large and sparse matrices $\mathbf{A}$ and $\mathbf{B}$. Sparsity or structures of $\mathbf{A}$ and $\mathbf{B}$ can be exploited in the matrix-vector multiplications. This enjoys the same feature as the Arnoldi process for generating an orthonormal basis of the standard Krylov subspace $\mathcal{K}_n$.

The **for**-loop in lines 6–10 is an orthogonalization procedure with respect to the $\{\mathbf{q}_i\}$ vectors. The vector sequence $\{\mathbf{p}_j\}$ is an auxiliary sequence. In section 3, we will present a modified version of the algorithm to remove the requirement of explicit reference of the sequence $\{\mathbf{p}_j\}$. This will reduce the memory requirements by almost half.

Algorithm 1 stops prematurely when the norm of $\mathbf{r}$ computed at line 12 vanishes at a certain step j. In this case, we encounter either *deflation* or *breakdown*. We delay the discussion of deflation and breakdown till the next section.

We now consider basic relations between quantities generated by the algorithm. If $\mathbf{Q}_n$ denotes the $N \times n$ matrix with column vectors $\mathbf{q}_1,\mathbf{q}_2,\ldots,\mathbf{q}_n$, $\mathbf{P}_n$ denotes the $N \times n$ matrix with column vectors $\mathbf{p}_1,\mathbf{p}_2,\ldots,\mathbf{p}_n$, and $\mathbf{T}_n$ denotes the $n \times n$ upper Hessenberg matrix with nonzero entries t_{ij} as defined in the algorithm, then the following relations hold:

$$\mathbf{A}\mathbf{Q}_n + \mathbf{B}\mathbf{P}_n = \mathbf{Q}_n\mathbf{T}_n + \mathbf{q}_{n+1}\mathbf{e}_n^{\mathrm{T}} t_{n+1\,n}, \tag{2.5}$$

$$\mathbf{Q}_n = \mathbf{P}_n\mathbf{T}_n + \mathbf{p}_{n+1}\mathbf{e}_n^{\mathrm{T}} t_{n+1\,n} \tag{2.6}$$

with the orthonormality of the vector sequence $\{\mathbf{q}_1,\mathbf{q}_2,\ldots,\mathbf{q}_n,\mathbf{q}_{n+1}\}$. Let $\widehat{\mathbf{T}}_n$ be an $(n+1)\times n$ upper Hessenberg matrix of the form $\widehat{\mathbf{T}}_n = \begin{bmatrix} \mathbf{T}_n \\ \mathbf{e}_n^{\mathrm{T}} t_{n+1\,n} \end{bmatrix}$. Then equations

(2.5) and (2.6) can be rewritten in the compact form

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_n \\ \mathbf{P}_n \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{n+1} \\ \mathbf{P}_{n+1} \end{bmatrix} \widehat{\mathbf{T}}_n. \tag{2.7}$$

This relation assembles the similarity between the SOAR procedure and the well-known Arnoldi procedure [1]. Let us recall the following Arnoldi procedure for generating an orthonormal basis $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ of the Krylov subspace $\mathcal{K}_n(\mathbf{H}; \mathbf{v})$, where $\mathbf{H}$ and $\mathbf{v}$ are defined in (2.4).

ALGORITHM 2. *Arnoldi procedure.*

1. $\mathbf{v}_1 = \mathbf{v}/\|\mathbf{v}\|_2$
2. **for** $j = 1, 2, \ldots, n$ **do**
3. $\quad \mathbf{r} = \mathbf{H}\mathbf{v}_j$
4. $\quad$ **for** $i = 1, 2, \ldots, j$ **do**
5. $\quad\quad h_{ij} = \mathbf{v}_i^{\mathrm{T}}\mathbf{r}$
6. $\quad\quad \mathbf{r} := \mathbf{r} - \mathbf{v}_i h_{ij}$
7. $\quad$ **end for**
8. $\quad h_{j+1\,j} = \|\mathbf{r}\|_2$
9. $\quad$ **if** $h_{j+1,j} = 0$, **breakdown**
10. $\quad \mathbf{v}_{j+1} = \mathbf{r}/h_{j+1\,j}$
11. **end for**

If $\mathbf{V}_n$ denotes the $2N \times n$ matrix with column vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ and $\mathbf{H}_n$ denotes the $n \times n$ Hessenberg matrix with nonzero entries h_{ij} as defined in the algorithm, then the Arnoldi procedure can be compactly expressed by the equation

$$\mathbf{H}\,\mathbf{V}_n = \mathbf{V}_n\mathbf{H}_n + \mathbf{v}_{n+1}\mathbf{e}_n^{\mathrm{T}} h_{n+1\,n}$$

or be cast in the form similar to (2.7),

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{V}_n = \mathbf{V}_{n+1}\widehat{\mathbf{H}}_n, \tag{2.8}$$

where $\mathbf{V}_{n+1} = \begin{bmatrix} \mathbf{V}_n & \mathbf{v}_{n+1} \end{bmatrix}$ is a $(2N) \times (n+1)$ orthonormal matrix, and $\widehat{\mathbf{H}}_n = \begin{bmatrix} \mathbf{H}_n \\ \mathbf{e}_n^{\mathrm{T}} h_{n+1\,n} \end{bmatrix}$ is a $(n+1) \times n$ upper Hessenberg matrix. By comparing (2.7) and (2.8), we see that the essential difference between the SOAR procedure and the Arnoldi procedure is that in SOAR, the nonzero elements t_{ij} of the $(n+1) \times n$ upper Hessenberg matrix $\widehat{\mathbf{T}}_n$ are chosen to enforce the orthonormality of the vectors $\{\mathbf{q}_j\}$ of dimension N, whereas in Arnoldi, the nonzero elements h_{ij} of $(n+1) \times n$ upper Hessenberg matrix $\widehat{\mathbf{H}}_n$ are determined to ensure the orthonormality of the vectors $\{\mathbf{v}_j\}$ of dimension $2N$. In the next section, we will further exploit the relationship between SOAR and Arnoldi to derive a revised version of the SOAR procedure, which remedies the deflation and saves half the memory requirement and floating point operations.

For the rest of this section, we prove that the vector sequence $\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n\}$ indeed is an orthonormal basis of the second-order Krylov subspace $\mathcal{G}_j(\mathbf{A}, \mathbf{B}; \mathbf{u})$. First, we have the following lemma, which reveals the connection between decomposition characteristics in (2.7) and (2.8) and a related Krylov subspace.

LEMMA 2.2. *Let* $\mathbf{A}$ *be an arbitrary* $n \times n$ *matrix. Let* $\mathbf{V}_{m+1} = \begin{bmatrix} \mathbf{V}_m & \mathbf{v}_{m+1} \end{bmatrix}$ *be an* $n \times (m+1)$ *rectangular matrix that satisfies*

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_{m+1}\widehat{\mathbf{H}}_m$$

for an $(m+1)\times m$ upper Hessenberg matrix $\widehat{\mathbf{H}}_m$. Then there is an upper triangular matrix $\mathbf{R}_m$ such that

$$\mathbf{V}_m\mathbf{R}_m = \begin{bmatrix}\mathbf{v}_1 & \mathbf{A}\mathbf{v}_1 & \cdots & \mathbf{A}^{m-1}\mathbf{v}_1\end{bmatrix}. \tag{2.9}$$

Furthermore, if the first $m-1$ subdiagonal elements of $\widehat{\mathbf{H}}_m$ are nonzero, then $\mathbf{R}_m$ is nonsingular and

$$\operatorname{span}\{\mathbf{V}_m\} = \mathcal{K}_m(\mathbf{A},\mathbf{v}_1). \tag{2.10}$$

Proof. We first prove (2.9) by induction on m. When $m=1$, (2.9) holds with $\mathbf{R}_1 = 1$. Assume that (2.9) holds for $m-1$. Then for m,

$$\begin{aligned}\begin{bmatrix}\mathbf{v}_1 & \mathbf{A}\mathbf{v}_1 & \cdots & \mathbf{A}^{m-1}\mathbf{v}_1\end{bmatrix} &= \begin{bmatrix}\mathbf{v}_1 & \mathbf{A}\begin{bmatrix}\mathbf{v}_1 & \mathbf{A}\mathbf{v}_1 & \cdots & \mathbf{A}^{m-2}\mathbf{v}_1\end{bmatrix}\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{v}_1 & \mathbf{A}\mathbf{V}_{m-1}\mathbf{R}_{m-1}\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{V}_m\mathbf{e}_1 & \mathbf{V}_m\widehat{\mathbf{H}}_{m-1}\mathbf{R}_{m-1}\end{bmatrix}\\ &= \mathbf{V}_m\begin{bmatrix}\mathbf{e}_1 & \widehat{\mathbf{H}}_{m-1}\mathbf{R}_{m-1}\end{bmatrix} \equiv \mathbf{V}_m\mathbf{R}_m.\end{aligned}$$

The fact of the upper triangularity of $\mathbf{R}_m$ is immediately followed by its definition. Furthermore, note that the diagonal elements of $\mathbf{R}_m$ are 1 and the products of the first $m-1$ subdiagonal elements of $\widehat{\mathbf{H}}_m$. Therefore, if these subdiagonal elements are nonzero, then $\mathbf{R}_m$ is nonsingular. Finally, (2.10) is established by (2.9) and the nonsingularity of $\mathbf{R}_m$. ∎

We note that in Lemma 2.2, the column vectors of $\mathbf{V}_n$ span the Krylov subspace $\mathcal{K}_n(\mathbf{A},\mathbf{v}_1)$ as long as (2.9) is satisfied and $\mathbf{R}_m$ is nonsingular. It is still true even when some of the columns of $\mathbf{V}_n$ are zero vectors. Lemma 2.2 can be viewed as a generalization of the second part of Theorem 1.1 in [21, p. 298]. We will apply this fact when we discuss the deflation in the SOAR procedure. We now prove that Algorithm 1 generates an orthonormal basis of the second-order Krylov subspace $\mathcal{G}_j(\mathbf{A},\mathbf{B};\mathbf{u})$.

THEOREM 2.3. *If $t_{j+1,j}\neq 0$ for $j\geq 1$ in Algorithm 1, then the vector sequence $\{\mathbf{q}_1,\mathbf{q}_2,\ldots,\mathbf{q}_j\}$ forms an orthonormal basis of the second-order Krylov subspace $\mathcal{G}_j(\mathbf{A},\mathbf{B};\mathbf{u})$, i.e.,*

$$\operatorname{span}\{\mathbf{Q}_j\} = \mathcal{G}_j(\mathbf{A},\mathbf{B};\mathbf{u}) \quad \text{for } j\geq 1 \tag{2.11}$$

and $\mathbf{q}_i^{\mathrm{T}}\mathbf{q}_k = 0$ if $i\neq k$ and $\mathbf{q}_i^{\mathrm{T}}\mathbf{q}_i = 1$ for $i,k = 1,2,\ldots,j$.

Proof. Equation (2.11) is established by the following sequence of equalities:

$$\begin{aligned}\mathcal{G}_j(\mathbf{A},\mathbf{B};\mathbf{r}_0) &= \operatorname{span}\{\mathbf{r}_0,\mathbf{r}_1,\ldots,\mathbf{r}_{j-1}\}\\ &= \operatorname{span}\left\{\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{r}_0 & \mathbf{r}_1 & \cdots & \mathbf{r}_{j-1}\\ \mathbf{0} & \mathbf{r}_0 & \cdots & \mathbf{r}_{j-2}\end{bmatrix}\right\}\\ &= \operatorname{span}\left\{\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{v}_1 & \mathbf{H}\mathbf{v}_1 & \ldots & \mathbf{H}^{j-1}\mathbf{v}_1\end{bmatrix}\right\} \quad \text{by (2.4)}\\ &= \operatorname{span}\left\{\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_j\\ \mathbf{P}_j\end{bmatrix}\mathbf{R}_j\right\} \quad \text{by (2.7) and Lemma 2.2}\\ &= \operatorname{span}\left\{\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_j\\ \mathbf{P}_j\end{bmatrix}\right\} \quad \text{by the assumption that } \mathbf{R}_j \text{ is nonsingular}\\ &= \operatorname{span}\{\mathbf{Q}_j\}.\end{aligned}$$

Finally, the orthogonality of the basis vectors $\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_j\}$ is directly obtained from the orthogonalization inner **for**-loop (lines 6–10) and the normalization step at line 13 of the SOAR procedure. ∎

3. An SOAR procedure. As we pointed out in the previous section, Algorithm 1 stops prematurely when the norm of $\mathbf{r}$ computed at line 12 vanishes at a certain step j. There are two possible explanations for this. One is that the vector sequence $\{\mathbf{r}_i\}$ for $i = 0, 1, \ldots, j-1$ is linearly dependent, but the double length vector sequence $\{[\mathbf{r}_i^{\mathrm{T}}\ \mathbf{r}_{i-1}^{\mathrm{T}}]^{\mathrm{T}}\}$ is linearly independent. We call this situation *deflation*. We will show that with a proper treatment, the SOAR procedure can continue. Another possible explanation is that both vector sequences $\{\mathbf{r}_i\}$ and $\{[\mathbf{r}_i^{\mathrm{T}}\ \mathbf{r}_{i-1}^{\mathrm{T}}]^{\mathrm{T}}\}$ are linearly dependent at a certain step j. In this situation, the SOAR procedure terminates. We call this *breakdown*.

The Arnoldi procedure (Algorithm 2) terminates when the norm of the vector $\mathbf{r}$ computed at line 9 vanishes at a certain step j. It happens when the vector sequence $\{\mathbf{H}^i\mathbf{v}\} = \{[\mathbf{r}_i^{\mathrm{T}}\ \mathbf{r}_{i-1}^{\mathrm{T}}]^{\mathrm{T}}\}$ for $i = 0, 1, \ldots, j-1$ is linearly dependent. This is known as the breakdown of the Arnoldi procedure.

In this section, we first discuss the deflation and then the breakdown. We will show the connection of breakdowns between the SOAR and Arnoldi procedures.

3.1. Deflation. We now present the following modified version of Algorithm 1, which remedies the deflation.

ALGORITHM 3. *SOAR procedure with deflation.*

```
 1.  q_1 = u/||u||_2
 2.  p_1 = 0
 3.  for j = 1, 2, ..., n do
 4.      r = Aq_j + Bp_j
 5.      s = q_j
 6.      for i = 1, 2, ..., j do
 7.          t_ij = q_i^T r
 8.          r := r − q_i t_ij
 9.          s := s − p_i t_ij
10.      end for
11.      t_{j+1 j} = ||r||_2
12.      if t_{j+1 j} = 0
13.          if s ∈ span{p_i | i : q_i = 0, 1 ≤ i ≤ j}
14.              breakdown
15           else    % deflation
16.              reset t_{j+1 j} = 1
17.              q_{j+1} = 0
18.              p_{j+1} = s
19.          end if
20.      else    % normal case
21.          q_{j+1} = r/t_{j+1 j}
22.          p_{j+1} = s/t_{j+1 j}
23.      end if
24.  end for
```

We note that in the modified SOAR procedure above, when deflation is detected (line 15), it simply takes $\mathbf{q}_{j+1} = \mathbf{0}$ and sets the scaling element $t_{j+1\,j}$ to a nonzero value (line 16). Then the procedure continues.

Without repeating the discussion in section 2, we state that quantities generated by Algorithm 3 hold the same relations as Algorithm 1, e.g., (2.7) is still true and the vector sequence $\{\mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_{n-1}\}$ still spans the second-order Krylov subspace $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$, except that some of the $\mathbf{q}$ vectors are zero vectors when deflations occur at the corresponding steps. The set of nonzero $\mathbf{q}$ vectors forms an orthonormal basis of $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$.

3.2. Breakdown. Let us discuss the situation where breakdown occurs. We have the following theorem.

THEOREM 3.1. *The SOAR procedure* (*Algorithm* 3) *with matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *and starting vector* $\mathbf{u}$ *breaks down at a certain step* j *if and only if the Arnoldi procedure with matrix* $\mathbf{H}$ *and starting vector* $\mathbf{v}$ *breaks down at the same step* j.

To prove Theorem 3.1, we need the following lemma.

LEMMA 3.2. *For a sequence of linearly independent vectors* $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ *with partition* $\mathbf{v}_i = \{[\,\mathbf{q}_i^{\mathrm{T}}\ \mathbf{p}_i^{\mathrm{T}}\,]^{\mathrm{T}}\}$, *if there exists a subsequence* $\{\mathbf{q}_{i_1}, \mathbf{q}_{i_2}, \ldots, \mathbf{q}_{i_k}\}$ *of the* $\mathbf{q}$ *vectors that are linearly independent and the remaining vectors are zeros,* $\mathbf{q}_{i_{k+1}} = \mathbf{q}_{i_{k+2}} = \cdots = \mathbf{q}_{i_n} = \mathbf{0}$, *then a vector* $\mathbf{v} = \{[\,\mathbf{0}\ \mathbf{p}^{\mathrm{T}}\,]^{\mathrm{T}}\} \in \mathrm{span}\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ *if and only if* $\mathbf{p} \in \mathrm{span}\{\mathbf{p}_{i_{k+1}}, \mathbf{p}_{i_{k+2}}, \ldots, \mathbf{p}_{i_n}\}$.

Proof. If $\mathbf{v} \in \mathrm{span}\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$, then there exist scalars α_i, such that $\mathbf{v} = \sum_{i=1}^{n} \alpha_i \mathbf{v}_i$. By the assumption that $\mathbf{v} = \{[\,\mathbf{0}\ \mathbf{p}^{\mathrm{T}}\,]^{\mathrm{T}}\} \in \mathrm{span}\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ and some zero vectors in the $\mathbf{q}$ vector sequence, we have $\mathbf{0} = \sum_{j=1}^{n} \alpha_j \mathbf{q}_j = \sum_{j=1}^{k} \alpha_{i_j} \mathbf{q}_{i_j}$. Since vectors $\mathbf{q}_{i_1}, \mathbf{q}_{i_2}, \ldots, \mathbf{q}_{i_k}$ are linearly independent, it yields that $\alpha_{i_j} = 0$ for $j = 1, 2, \ldots, k$. Hence $\mathbf{v} = \sum_{j=k+1}^{n} \alpha_{i_j} \mathbf{v}_{i_j}$, which means that $\mathbf{p} = \sum_{j=k+1}^{n} \alpha_{i_j} \mathbf{p}_{i_j}$ or, equivalently, $\mathbf{p} \in \mathrm{span}\{\mathbf{p}_{i_{k+1}}, \mathbf{p}_{i_{k+2}}, \ldots, \mathbf{p}_{i_n}\}$. ∎

Proof of Theorem 3.1. Let us first consider that the Arnoldi procedure breaks down at a certain step j. This implies that

$$\dim(\mathcal{K}_n(\mathbf{H}, \mathbf{v})) = j \quad \text{and} \quad \mathbf{H}^n \mathbf{v} \in \mathcal{K}_j(\mathbf{H}, \mathbf{v}) \quad \text{for } n \geq j \tag{3.1}$$

From (2.7) and Lemma 2.2, we have

$$\mathrm{span}\left\{ \begin{bmatrix} \mathbf{Q}_j \\ \mathbf{P}_j \end{bmatrix} \right\} = \mathcal{K}_j(\mathbf{H}, \mathbf{v}).$$

Since $\dim(\mathcal{K}_j(\mathbf{H}, \mathbf{v})) = j$, $\left[\begin{smallmatrix} \mathbf{Q}_j \\ \mathbf{P}_j \end{smallmatrix}\right]$ is full column rank. By Lemma 2.2 again and (3.1), we have

$$\begin{bmatrix} \mathbf{r} \\ \mathbf{s} \end{bmatrix} \in \mathrm{span}\left\{ \begin{bmatrix} \mathbf{Q}_j \\ \mathbf{P}_j \end{bmatrix} \right\}. \tag{3.2}$$

We now show that $\mathbf{r} = \mathbf{0}$ (at line 11 of Algorithm 3). Suppose $\mathbf{r} \neq \mathbf{0}$. Since $\mathbf{r}^{\mathrm{T}} \mathbf{q}_i = 0$ for $i = 1, 2, \ldots, j$, it implies that

$$\mathbf{r} \notin \mathrm{span}\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_j\},$$

which indicates that

$$\begin{bmatrix} \mathbf{r} \\ \mathbf{s} \end{bmatrix} \notin \mathrm{span}\left\{ \begin{bmatrix} \mathbf{Q}_j \\ \mathbf{P}_j \end{bmatrix} \right\}.$$

This contradicts (3.2). Therefore $\mathbf{r} = \mathbf{0}$. Thus Algorithm 3 proceeds to execute line 13. By (3.2) and Lemma 3.2, we have

$$\mathbf{s} \in \mathrm{span}\{\mathbf{p}_i \mid i : \mathbf{q}_i = \mathbf{0}, 1 \leq i \leq j\}.$$

Therefore, Algorithm 3 also breaks down (line 14 of Algorithm 3).

Conversely, if Algorithm 3 breaks down at a certain step j, then

$$\begin{bmatrix} \mathbf{r} \\ \mathbf{s} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{s} \end{bmatrix} \in \operatorname{span}\left\{ \begin{bmatrix} \mathbf{Q}_j \\ \mathbf{P}_j \end{bmatrix} \right\}. \tag{3.3}$$

Note that (2.7) still holds with the choice of $t_{j+1\,j} = 1$. Thus by Lemma 2.2, we have

$$\operatorname{span}\left\{ \begin{bmatrix} \mathbf{Q}_j \\ \mathbf{P}_j \end{bmatrix} \right\} = \mathcal{K}_j(\mathbf{H}, \mathbf{v}) \quad \text{and} \quad \operatorname{span}\left\{ \begin{bmatrix} \mathbf{Q}_{j+1} \\ \mathbf{P}_{j+1} \end{bmatrix} \right\} = \mathcal{K}_{j+1}(\mathbf{H}, \mathbf{v}).$$

On the other hand, by induction, we can show that after $j-1$ steps in Algorithm 3, we have

$$\operatorname{rank}\left(\begin{bmatrix} \mathbf{Q}_j \\ \mathbf{P}_j \end{bmatrix} \right) = j. \tag{3.4}$$

Combining (3.3) and (3.4), it yields that $\dim(\mathcal{K}_j(\mathbf{H}, \mathbf{v})) = j$ and $\mathcal{K}_j(\mathbf{H}, \mathbf{v}) = \mathcal{K}_{j+1}(\mathbf{H}, \mathbf{v})$. These two conditions ensure that the Arnoldi procedure breaks down at the same step j. ∎

In the Arnoldi procedure, when breakdown occurs, it indicates that the Krylov subspace $\mathcal{K}_j(\mathbf{H}, \mathbf{v})$ is an invariant subspace of $\mathbf{H}$, and the vector sequence $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_j\}$ is an orthonormal basis of the subspace. It is regarded as a lucky breakdown. For the SOAR procedure (Algorithm 3), by (2.7) we know that the column vectors of the $2N \times j$ matrix $\left[\begin{smallmatrix} \mathbf{Q}_j \\ \mathbf{P}_j \end{smallmatrix}\right]$ also span an invariant subspace of $\mathbf{H}$, but it is not an orthonormal basis.

3.3. An SOAR procedure. Now we further exploit the relations in Algorithm 3 to derive a new version, which avoids the explicit references and updates of the $\mathbf{p}$ vectors at lines 9 and 22. The resulting algorithm reduces memory requirement by almost half.

First, by (2.6) and noting that $\mathbf{p}_1 = \mathbf{0}$, we have

$$\mathbf{Q}_n = \mathbf{P}_{n+1}\widehat{\mathbf{T}}_n = \mathbf{P}_{n+1}(:, 2 : n+1) \cdot \widehat{\mathbf{T}}_n(2 : n+1, 1 : n).$$

Then (2.5) can be rewritten as

$$\mathbf{A}\mathbf{Q}_n + \mathbf{B}\mathbf{Q}_n\mathbf{S}_n = \mathbf{Q}_n\mathbf{T}_n + \mathbf{q}_{n+1}\mathbf{e}_n^{\mathrm{T}} t_{n+1\,n}, \tag{3.5}$$

where $\mathbf{S}_n$ is an $n \times n$ strictly upper triangular matrix of the form

$$\mathbf{S}_n = \begin{bmatrix} \mathbf{0} & \widehat{\mathbf{T}}_n(2 : n, 1 : n-1)^{-1} \\ 0 & \mathbf{0} \end{bmatrix}.$$

Equation (3.5) suggests a method for computing vector $\mathbf{q}_{j+1}$ from $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_j$. This leads to the following algorithm, which needs only about a half of the memory and floating point operations of Algorithm 3.

ALGORITHM 4. *SOAR procedure with deflation and memory saving.*

1. $\mathbf{q}_1 = \mathbf{u}/\|\mathbf{u}\|_2$
2. $\mathbf{f} = \mathbf{0}$
3. **for** $j = 1, 2, \ldots, n$ **do**
4. $\quad \mathbf{r} = \mathbf{A}\mathbf{q}_j + \mathbf{B}\mathbf{f}$
5. $\quad$ **for** $i = 1, 2, \ldots, j$ **do**
6. $\quad\quad t_{ij} = \mathbf{q}_i^{\mathrm{T}}\mathbf{r}$
7. $\quad\quad \mathbf{r} := \mathbf{r} - \mathbf{q}_i t_{ij}$
8. $\quad$ **end for**
9. $\quad t_{j+1\,j} = \|\mathbf{r}\|_2$
10. $\quad$ **if** $t_{j+1\,j} \neq 0$,
11. $\quad\quad \mathbf{q}_{j+1} := \mathbf{r}/t_{j+1\,j}$
12. $\quad\quad \mathbf{f} = \mathbf{Q}_j \widehat{\mathbf{T}}(2 : j+1, 1 : j)^{-1}\mathbf{e}_j$
13. $\quad$ **else**
14. $\quad\quad$ *reset* $t_{j+1\,j} = 1$
15. $\quad\quad \mathbf{q}_{j+1} = \mathbf{0}$
16. $\quad\quad \mathbf{f} = \mathbf{Q}_j \widehat{\mathbf{T}}(2 : j+1, 1 : j)^{-1}\mathbf{e}_j$
17. $\quad\quad$ *save* $\mathbf{f}$ *and check deflation and breakdown*
18. $\quad$ **end if**
19. **end for**

Note that at line 17 of the algorithm above, if $\mathbf{f}$ belongs to the subspace spanned by previously saved $\mathbf{f}$ vectors, then the algorithm encounters breakdown and terminates. Otherwise, there is a deflation at step j; after setting $t_{j+1\,j}$ to 1 or any nonzero constant, the algorithm continues. Those saved $\mathbf{f}$ vectors are the $\mathbf{p}_i$ vectors corresponding to the vector $\mathbf{q}_i = \mathbf{0}$ in Algorithm 3. To check whether $\mathbf{f}$ is in the subspace spanned by the previously saved $\mathbf{f}$, we can use a modified Gram–Schmidt procedure [21]. It is not necessary to use extra storage to save those $\mathbf{f}$ vectors. They can be stored at the columns of $\mathbf{Q}_n$ where the corresponding $\mathbf{q}_i = \mathbf{0}$.

4. A projection method applied directly to the QEP. In this section, we apply the concept of the second-order Krylov subspace and its orthonormal basis generated by the SOAR procedure to develop a projection technique to solve the QEP (1.2). We follow the orthogonal Rayleigh–Ritz approximation procedure to derive a method which approximates a large-scale QEP by a small-scale QEP.

Following the standard derivation, to apply the Rayleigh–Ritz approximation technique based on the subspace $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$ with $\mathbf{A} = -\mathbf{M}^{-1}\mathbf{D}$ and $\mathbf{B} = -\mathbf{M}^{-1}\mathbf{K}$, we seek an approximate eigenpair $(\theta, \mathbf{z})$, where $\theta \in \mathcal{C}$ and $\mathbf{z} \in \mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$, by imposing the following orthogonal condition, also called the Galerkin condition:

$$(\theta^2\mathbf{M} + \theta\mathbf{D} + \mathbf{K})\mathbf{z} \;\perp\; \mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$$

or, equivalently,

$$\mathbf{v}^{\mathrm{T}}(\theta^2\mathbf{M} + \theta\mathbf{D} + \mathbf{K})\mathbf{z} = 0 \quad \text{for all } \mathbf{v} \in \mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u}). \tag{4.1}$$

Since $\mathbf{z} \in \mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$, it can be written as

$$\mathbf{z} = \mathbf{Q}_m\mathbf{g}, \tag{4.2}$$

where the $N \times m$ matrix $\mathbf{Q}_m$ is an orthonormal basis of $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$ generated by the SOAR procedure (Algorithm 4), and $\mathbf{g}$ is an m vector and $m \leq n$. When there are

deflations, $m < n$. By (4.1) and (4.2), it yields that θ and $\mathbf{g}$ must satisfy the reduced QEP:

$$(\theta^2\mathbf{M}_m + \theta\mathbf{D}_m + \mathbf{K}_m)\mathbf{g} = \mathbf{0} \tag{4.3}$$

with

$$\mathbf{M}_m = \mathbf{Q}_m^{\mathrm{T}}\mathbf{M}\mathbf{Q}_m, \quad \mathbf{D}_m = \mathbf{Q}_m^{\mathrm{T}}\mathbf{D}\mathbf{Q}_m, \quad \mathbf{K}_m = \mathbf{Q}_m^{\mathrm{T}}\mathbf{K}\mathbf{Q}_m. \tag{4.4}$$

The eigenpairs $(\theta, \mathbf{g})$ of (4.3) define the *Ritz pairs* $(\theta,\ \mathbf{z})$. The Ritz pairs are approximate eigenpairs of the QEP (1.2). The accuracy of the approximate eigenpairs $(\theta, \mathbf{z})$ can be assessed by the norms of the residual vectors $(\theta^2\mathbf{M} + \theta\mathbf{D} + \mathbf{K})\mathbf{z}$.

We note that by explicitly formulating the matrices $\mathbf{M}_m$, $\mathbf{D}_m$, and $\mathbf{K}_m$, essential structures of $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ are preserved. For example, if $\mathbf{M}$ is symmetric positive definite, so is $\mathbf{M}_m$. As a result, essential spectral properties of the QEP will be preserved. For example, if the QEP is a gyroscopic dynamical system in which $\mathbf{M}$ and $\mathbf{K}$ are symmetric, one of them is positive definite, and $\mathbf{D}$ is skew-symmetric, then the reduced QEP is also a gyroscopic system. It is known that in this case, the eigenvalues λ are symmetrically placed with respect to both the real and imaginary axes [10]. Such a spectral property will be preserved in the reduced QEP.

The following algorithm is a high-level description of the Rayleigh–Ritz projection procedure based on $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$ for solving the QEP (1.2) directly.

ALGORITHM 5. *SOAR method for solving the QEP directly.*

1. *Run the SOAR procedure* (*Algorithm* 4) *with* $\mathbf{A} = -\mathbf{M}^{-1}\mathbf{D}$ *and* $\mathbf{B} = -\mathbf{M}^{-1}\mathbf{K}$ *and a starting vector* $\mathbf{u}$ *to generate an* $N \times m$ *orthogonal matrix* $\mathbf{Q}_m$ *whose columns span an orthonormal basis of* $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$.
2. *Compute* $\mathbf{M}_m$, $\mathbf{D}_m$, *and* $\mathbf{K}_m$ *as defined in* (4.4).
3. *Solve the reduced QEP* (4.3) *for* $(\theta, \mathbf{g})$ *and obtain the Ritz pairs* $(\theta, \mathbf{z})$, *where* $\mathbf{z} = \mathbf{Q}_m\mathbf{g}/\|\mathbf{Q}_m\mathbf{g}\|_2$.
4. *Test the accuracy of Ritz pairs* $(\theta,\ \mathbf{z})$ *as approximate eigenvalues and eigenvectors of the QEP* (1.2) *by the relative norms of residual vectors:*

$$\frac{\|(\theta^2\mathbf{M} + \theta\mathbf{D} + \mathbf{K})\mathbf{z}\|_2}{|\theta|^2\|\mathbf{M}\|_1 + |\theta|\|\mathbf{D}\|_1 + \|\mathbf{K}\|_1}. \tag{4.5}$$

A few remarks are in order:

- At step 1, the matrix-vector product operations $-\mathbf{M}^{-1}\mathbf{D}\mathbf{u}$ and $-\mathbf{M}^{-1}\mathbf{K}\mathbf{u}$ for an arbitrary vector $\mathbf{u}$ must be provided to run the SOAR procedure (Algorithm 4). A factorized form of $\mathbf{M}$, such as the LU factorization, should be made available outside of the first **for**-loop of Algorithm 4 for computational efficiency.
- At step 2, the orthonormal basis matrix $\mathbf{Q}_m$ computed in step 1 is used to explicitly compute the projection matrices $\mathbf{M}_n$, $\mathbf{D}_n$, and $\mathbf{K}_n$. This can be done by using matrix-vector product operations $\mathbf{M}\mathbf{q}$, $\mathbf{D}\mathbf{q}$, and $\mathbf{K}\mathbf{q}$ for an arbitrary vector $\mathbf{q}$. This is an overhead comparison of the method based on the Arnoldi procedure, in which the projection of the matrix is obtained as a by-product without any additional cost (see Algorithm 6 below). This overhead could be significant in some applications. However, this is a numerically better way to use the computed orthonormal basis $\mathbf{Q}_m$ since we can preserve the structures of coefficient matrices as we discussed early. Structure preservation often outweighs the extra cost of floating point operations in the modern

computing environment. For the numerical examples, presented in the next section, we observed that this step takes a small fraction of the total work, due to extreme sparsity of the matrices $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ in practical problems we encountered. The bottleneck of computational costs is often associated with the matrix-vector multiplication operations involving $\mathbf{M}^{-1}$ at step 1.

- At step 3, to solve the small QEP (4.3), we transform it to a generalized eigenvalue problem in the form of (1.3) and then use a dense matrix method, such as the QZ algorithm [5, 6], to find all eigenvalues and eigenvectors $(\theta, \mathbf{g})$ of the small QEP.
- At step 4, we use the relative residual norms (4.5) as the accuracy assessment to indicate the backward errors of the approximate eigenpairs $(\theta, \mathbf{z})$. The discussion of forward errors of approximate eigenvalues and eigenvectors is beyond the scope of this report; the interested reader is referred to [11, 23, 24].

Let us review the basic Arnoldi method for solving the QEP (1.2) based on linearization (1.3). At this stage of our study, we are concerned only with the fundamental properties and behaviors of the SOAR method. It is implemented in a straightforward way as outlined in Algorithm 5. Therefore, we will compare the SOAR method with the following simple implementation of the Arnoldi method for solving the QEP via linearization.

ALGORITHM 6. *Basic Arnoldi method for linearized QEP.*

1. *Transform the QEP* (1.2) *into the equivalent generalized eigenvalue problem* (1.3).
2. *Run the Arnoldi procedure* (*Algorithm* 2) *with the matrix* $\mathbf{H} = \mathbf{G}^{-1}\mathbf{C}$ *and the vector* $\mathbf{v} = [\,\mathbf{u}^{\mathrm{T}}\ \mathbf{0}\,]^{\mathrm{T}}$ *to generate an orthonormal basis* $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ *of the Krylov subspace* $\mathcal{K}_n(\mathbf{H}; \mathbf{v})$. *Let* $\mathbf{V}_n = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$.
3. *Solve the reduced eigenvalue problem*

$$(\mathbf{V}_n^{\mathrm{T}}\mathbf{H}\mathbf{V}_n)\mathbf{t} = \theta\mathbf{t}$$

and obtain the Ritz pairs $(\theta, \mathbf{y})$ *of the eigenvalue problem of the single matrix* $\mathbf{H}$, *where* $\mathbf{y} = \mathbf{V}_n\mathbf{t}$. *Note that by* (2.8), $\mathbf{V}_n^{\mathrm{T}}\mathbf{H}\mathbf{V}_n = \mathbf{H}_n(1:n, 1:n)$ *is an* $n \times n$ *upper Hessenberg matrix returned directly from the Arnoldi procedure without additional cost.*

4. *Extract the approximate eigenpairs* $(\theta,\ \mathbf{z})$ *of the QEP* (1.2) *and test their accuracy by the residual norms as described in* (4.5), *where* $\mathbf{z} = \mathbf{y}(N+1 : 2N)/\|\mathbf{y}(N+1:2N)\|_2$.

Finally, we discuss a hybrid method of the SOAR method (Algorithm 5) and the Arnoldi method (Algorithm 6) to solve the QEP directly. This method provides a good verification for the SOAR method. Let $\mathbf{K}_n$ denote the matrix of the explicit Krylov basis of $\mathcal{K}_n(\mathbf{H}, \mathbf{v})$:

$$\mathbf{K}_n = [\ \mathbf{v}\ \ \mathbf{H}\mathbf{v}\ \ \mathbf{H}^2\mathbf{v}\ \ \cdots\ \ \mathbf{H}^{n-1}\mathbf{v}\ ].$$

Then it is well known (for example, see [21, section 5.1]) that $\mathbf{V}_n$, generated by the Arnoldi procedure with $\mathbf{H}$ and $\mathbf{v}$, is the Q-factor of the QR factorization of $\mathbf{K}_n$:

$$\mathbf{K}_n = \mathbf{V}_n\mathbf{R}_n.$$

By (2.4), the equation above can be written in the form

$$\begin{bmatrix} \mathbf{r}_0 & \mathbf{r}_1 & \cdots & \mathbf{r}_{n-1} \\ \mathbf{0} & \mathbf{r}_0 & \cdots & \mathbf{r}_{n-2} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_n^{(1)} \\ \mathbf{V}_n^{(2)} \end{bmatrix} \mathbf{R}_n,$$

where $\mathbf{V}_n$ is partitioned into a 2×1 block matrix with $N \times n$ subblocks $\mathbf{V}_n^{(1)}$ and $\mathbf{V}_n^{(2)}$. From the first N rows of the previous equation, we have

$$\begin{bmatrix} \mathbf{r}_0 & \mathbf{r}_1 & \cdots & \mathbf{r}_{n-1} \end{bmatrix} = \mathbf{V}_n^{(1)} \mathbf{R}_n. \tag{4.6}$$

Hence, we have

$$\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u}) = \operatorname{span}\{\mathbf{V}_n^{(1)}\}.$$

Therefore, an alternative way to generate an orthonormal basis of $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$ is to first run the Arnoldi procedure with $2N \times 2N$ matrix $\mathbf{H}$ and starting vector $\mathbf{v} = [\, \mathbf{u}^{\mathrm{T}} \;\; \mathbf{0}]^{\mathrm{T}}$, then orthonormalize the first block $\mathbf{V}_n^{(1)}$ of $\mathbf{V}_n$ to obtain an orthonormal basis of the projection subspace $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$. This method provides a good verification for the SOAR method, although it is expensive in terms of memory and computational requirements. For numerical results presented in the next section, we observed that the convergence rate and behaviors of this method and the SOAR method are essentially the same.

5. Numerical examples. In this section, we present numerical examples to demonstrate the promises of the SOAR method (Algorithm 5) for solving the QEP (1.2). Following the discussion presented in the previous sections, we focus on the illustration of the fundamental properties of the SOAR method in terms of the following two aspects:

1. The convergence behaviors of the SOAR method applied directly to the QEP are generally comparable to the Arnoldi method applied to the linearized QEP. Specifically,
   (a) eigenvalues with the largest magnitude converge first;
   (b) the convergence rate of the SOAR method is at least as fast as the Arnoldi method.
2. The SOAR method preserves the essential structures of the QEP, such as symmetry and positive definiteness in coefficient matrices $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$. As a result, we should expect the preservation of spectral properties of the large QEP (1.2) in the reduced QEP (4.3).

In the following examples, the starting vector $\mathbf{u}$ of the SOAR method is chosen as a vector with all components equal to 1. $\mathbf{v} = [\mathbf{u}^{\mathrm{T}} \;\; \mathbf{0}]^{\mathrm{T}}$ is used as the starting vector of the Arnoldi-based methods (Algorithms 6 and 7). The so-called *exact* eigenvalues of the QEP are computed by the dense method, namely, the QZ method for computing all eigenvalues and eigenvectors of the generalized eigenvalue problem (1.3). The deflation and breakdown thresholds are set to be the same, namely, 10^{-10}. In fact, with this threshold, deflation and breakdown were detected only in Example 1.

Example 1. This example shows the deflation and breakdown phenomena in the SOAR procedure (Algorithm 4). The matrices $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ are from the modeling of a simple vibrating spring-mass system with damping in linear connection [5, 9]. $\mathbf{M}$ and $\mathbf{D}$ are diagonal matrices, and $\mathbf{K}$ is tridiagonal. For this particular run, we choose 50×50 matrices, where $\mathbf{M} = 0.1 \times \mathbf{I}$, $\mathbf{D} = \mathbf{I}$, and

$$\mathbf{K} = \begin{bmatrix} 0.2 & -0.1 & & & \\ -0.1 & 0.2 & -0.1 & & \\ & \ddots & \ddots & \ddots & \\ & & -0.1 & 0.2 & -0.1 \\ & & & -0.1 & 0.1 \end{bmatrix}.$$

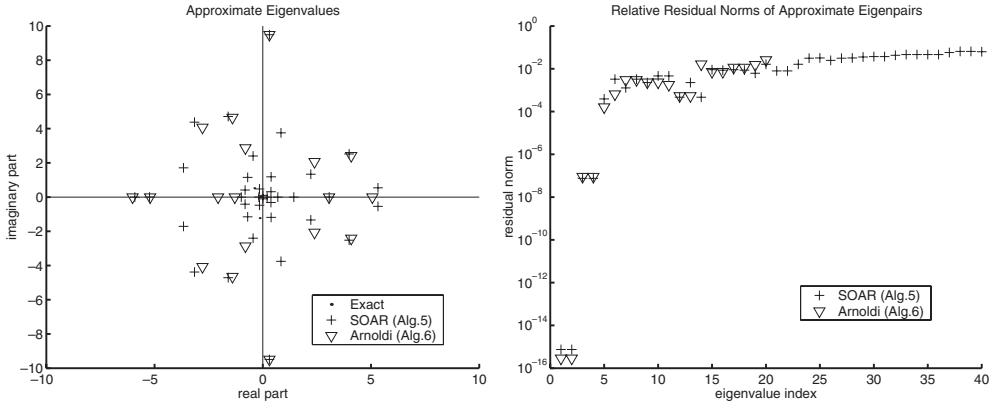


FIG. 5.1. *Random nonsymmetric QEP; exact and approximate eigenvalues* (*left*)*, and relative residual norms* (*right*) (*Example* 2).

This example illustrates the following two main issues:

1. Deflation occurs at every even step of the SOAR procedure, i.e., $\mathbf{q}_j = 0$ for all even number j.
2. Suppose the starting vector $\mathbf{u}$ is chosen as a linear combination of κ eigenvectors of the matrix $\mathbf{K}$ corresponding to the κ eigenvalues closest to 0. For $\kappa = 1, 2, 3$, both the SOAR procedure (Algorithm 4) and the Arnoldi procedure (Algorithm 2) break down at steps $j = 2\kappa$. However, for large κ, breakdown has not been detected due to numerical noises.

Example 2. The purpose of this example is to show that the convergence behaviors of the SOAR and Arnoldi methods are generally the same for a "general" QEP. Let $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ be 200×200 random nonsymmetric matrices. Elements of these matrices are chosen from a normal distribution with mean zero, variance one, and standard deviation one. The left plot of Figure 5.1 shows the partial approximate eigenvalues computed by two methods with the reduced dimension $n = 20$. The right plot of Figure 5.1 shows the relative residual norms. This example shows that the convergence behaviors of the two methods are essentially the same, as we expected.

Example 3. As in Example 2, this example is to show that the convergence rates of the SOAR and Arnoldi methods are comparable. However, only the SOAR method preserves the essential properties of the QEP. Specifically, $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ are chosen as 200×200 random matrices with the elements chosen from a normal distribution with mean zero, variance one, and standard deviation one. Furthermore, $\mathbf{M}$ is symmetric positive definite, $\mathbf{D}$ is skew-symmetric, and $\mathbf{K}$ is symmetric negative definite, as one encounters in a gyroscopic dynamical system. The gyroscopic system is a widely studied system. There are many interesting properties associated with such a system. For example, it is known that the distribution of the eigenvalues of the system in the complex plane is symmetric with respect to both the real and imaginary axes. The left plot of Figure 5.2 shows the approximate eigenvalues computed by two algorithms with $n = 20$. The right plot of Figure 5.2 shows the relative residual norms. This example shows that the SOAR method (Algorithm 5) preserves the gyroscopic spectral property. Furthermore, the residual norms indicate that the SOAR method has a slightly better convergence rate.

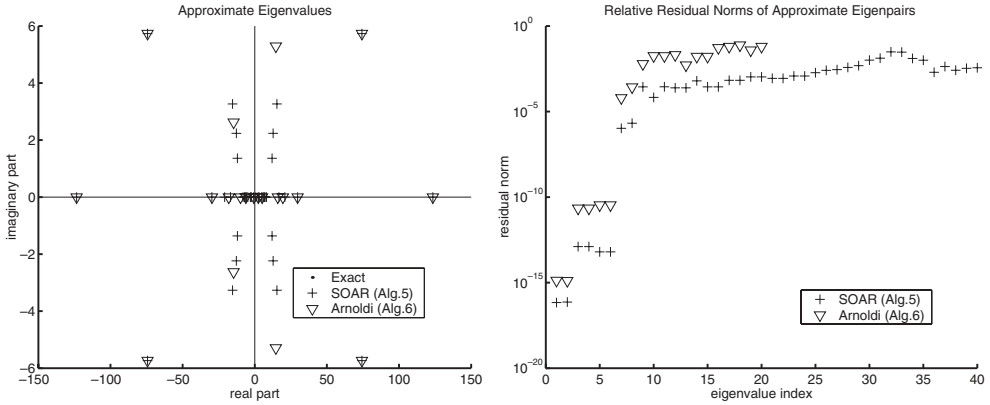


FIG. 5.2. *Random gyroscopic QEP; exact and approximate eigenvalues* (*left*) *and relative residual norms* (*right*) (*Example* 3).

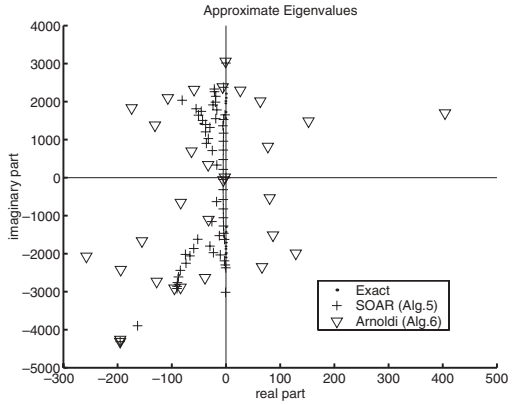


FIG. 5.3. *Acoustic QEP; exact and approximate eigenvalues* (*Example* 4).

Example 4. This is a QEP encountered in modeling the propagation of sound waves in a room in which one wall was made of a sound-absorbing material. This is a scaled-down version of the test problem as presented in [18]. The matrices $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ are of order 1331. Furthermore, $\mathbf{M}$ and $\mathbf{K}$ are real symmetric positive definite, and $\mathbf{D}$ is complex non-Hermitian. The largest magnitude eigenvalue computed by the standard dense matrix method (for all eigenvalues) and by the SOAR and Arnoldi methods with $n = 30$ are

$$\lambda_{\max} = -1.952652244810165 \times 10^2 - 4.314162072894026 \times 10^3\mathtt{i} \text{ (“exact”)},$$
$$\lambda^{\mathrm{S}}_{\max} = -\underline{1.952652244}09287 \times 10^2 - \underline{4.314162072894}454 \times 10^3\mathtt{i} \text{ (SOAR)},$$
$$\lambda^{\mathrm{A}}_{\max} = -\underline{1.952652}250694968 \times 10^2 - \underline{4.314162072}541710 \times 10^3\mathtt{i} \text{ (Arnoldi)}.$$

We observed that both the SOAR and Arnoldi methods converge to the largest magnitude eigenvalue first. The relative errors are $|\lambda^{\mathrm{S}}_{\max} - \lambda_{\max}|/|\lambda_{\max}| = 2.64 \times 10^{-12}$ and $|\lambda^{\mathrm{A}}_{\max} - \lambda_{\max}|/|\lambda_{\max}| = 1.95 \times 10^{-8}$, respectively. The largest magnitude eigenvalues produced by the SOAR method (Algorithm 5) are more accurate than the Arnoldi method (Algorithm 6). Furthermore, Figure 5.3 shows that all eigenvalues of the

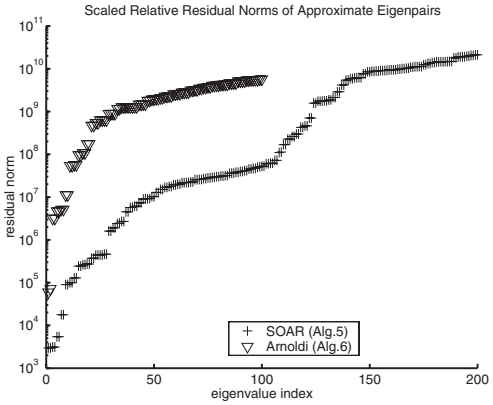


FIG. 5.4. *Scaled relative residual norms of Example* 5.

original QEP are distributed in the left half of the complex plane, known as stable eigenvalues. The reduced QEP by the SOAR method inherits such a property in the process of approximation. On the other hand, the linearized QEP used in the Arnoldi method loses this important property.

Example 5. This is a QEP problem from the NASTRAN simulation of a fluid-structure coupling cylinder model with both acoustic elements and structure elements. The order of the matrices $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ is $N = 3600$. The following table is a profile of other properties of the matrix triplet. The last column is an estimated lower bound for the 1-norm condition number using MATLAB's `condest` function.

	Nonzeros	Symmetry	Pos.def.	1-norm	Cond.est
$\mathbf{M}$	5521	yes	no	36.00	Inf
$\mathbf{D}$	19570	yes	no	1.025	Inf
$\mathbf{K}$	59062	yes	no	2.19×10^{12}	8.42×10^{16}

We solved the shift-and-invert QEP

$$(\mu^2\widehat{\mathbf{M}} + \mu\widehat{\mathbf{D}} + \widehat{\mathbf{K}})\mathbf{x} = \mathbf{0}, \tag{5.1}$$

where $\mu = 1/(\lambda - \sigma)$, $\widehat{\mathbf{M}} = \sigma^2\mathbf{M} + \sigma\mathbf{D} + \mathbf{K}$, $\widehat{\mathbf{D}} = \mathbf{D} + 2\sigma\mathbf{M}$, and $\widehat{\mathbf{K}} = \mathbf{M}$. The largest (in modulus) eigenvalue μ approximates the eigenvalues λ of the original QEP closest to the shift σ. These eigenvalues are given by $\sigma + 1/\mu$. With the shift $\sigma = 10^4$, a lower bound for the 1-norm condition number of the matrix $\widehat{\mathbf{M}}$ is 4.09×10^{13}. Figure 5.4 reports the scaled relative residual norms of the two methods with the subspace dimension $n = 100$. The scaled relative residual norm for an approximate eigenpair $(\theta, \mathbf{z})$ is defined by

$$\frac{\|(\theta^2\mathbf{M} + \theta\mathbf{D} + \mathbf{K})\mathbf{z}\|_2}{\epsilon\,(|\theta|^2\|\mathbf{M}\|_1 + |\theta|\|\mathbf{D}\|_1 + \|\mathbf{K}\|_1)},$$

where ϵ is the machine precision, which is at the order of 10^{-16} in double precision arithmetic. Since the norm of the matrix $\mathbf{M}$ is at the order of 10^{12}, it is better to show the scaled relative residual norm. To machine precision backward accuracy, the scaled relative residual norm should be about one.

Example 6. This final example arises from a finite element analysis of dissipative acoustics [4, 24]. Our matrix data for the associated algebraic quadratic eigenvalue

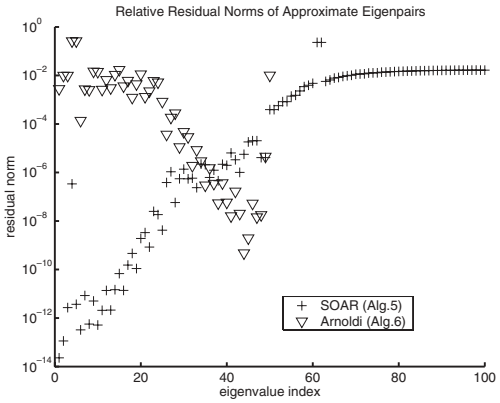


FIG. 5.5. *Relative residual norms of Example* 6.

problem are from [7]. The dimension of the QEP is $N = 9168$. Matrix $\mathbf{M}$ is symmetric positive definite, and matrices $\mathbf{D}$ and $\mathbf{K}$ are symmetric positive semidefinite. As described in [7], to find the eigenvalues of interest, we solve the shift-and-invert QEP (5.1) with the shift $\sigma = -253$. Figure 5.5 shows the relative residual norms for the approximated eigenpairs computed by the SOAR and Arnoldi methods with $n = 50$. We observe that SOAR converges faster than Arnoldi. By the Krylov-type subspace method proposed in [7], it is reported that with the number of iterations $n = 250$ to 300, three approximated eigenpairs converge with relative residual norms less than 10^{-12}. By contrast, the SOAR method delivers twice as many approximated eigenpairs with the same accuracy but only uses one-fifth of the number of iterations.

6. Discussion and future work. The primary purpose of this paper is to present the basic concept of the second-order Krylov subspace $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$ and its straightforward application for solving a large-scale QEP. There are many issues to examine. Foremost, one can ask whether the subspace $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$ is a better projection subspace to work with for an iterative solution of the QEP. A partial answer is based on the following observation. Let $\mathbf{A} = -\mathbf{M}^{-1}\mathbf{D}$ and $\mathbf{B} = -\mathbf{M}^{-1}\mathbf{K}$; then the QEP (1.2) is equivalent to the QEP

$$(\lambda^2\mathbf{I} - \lambda\mathbf{A} - \mathbf{B})\mathbf{x} = \mathbf{0}, \tag{6.1}$$

which can be written as the linear eigenvalue problem

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \lambda\mathbf{x} \\ \mathbf{x} \end{bmatrix} = \lambda \begin{bmatrix} \lambda\mathbf{x} \\ \mathbf{x} \end{bmatrix}. \tag{6.2}$$

In the Arnoldi basis $\mathbf{V}_n$ of the Krylov subspace $\mathcal{K}_n$, the coefficient matrix of (6.2) is represented by an upper Hessenberg matrix of order n,

$$\mathbf{V}_n^{\mathrm{T}} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{V}_n = \mathbf{H}_n. \tag{6.3}$$

On the other hand, using an orthonormal basis $\mathbf{Q}_n$ of the second-order Krylov subspace $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$, the coefficient matrix of (6.2) is represented by a 2×2 block matrix of order $2n$,

$$\begin{bmatrix} \mathbf{Q}_n^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_n^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_n \end{bmatrix} = \begin{bmatrix} \mathbf{A}_n & \mathbf{B}_n \\ \mathbf{I}_n & \mathbf{0} \end{bmatrix}. \tag{6.4}$$

It can be shown that the subspace spanned by the columns of $\mathbf{V}_n$ can be embedded into the subspace spanned by the columns of the 2×2 block diagonal matrix $\mathrm{diag}(\mathbf{Q}_n, \mathbf{Q}_n)$, namely,

$$\mathrm{span}\{\mathbf{V}_n\} \subset \mathrm{span}\left\{ \begin{bmatrix} \mathbf{Q}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_n \end{bmatrix} \right\}.$$

Therefore, the $2n \times 2n$ block matrix in (6.4) should deliver at least as many good approximations of eigenpairs as the $n \times n$ Hessenberg matrix $\mathbf{H}_n$ does.

We note that the explicit triangular inversion in the SOAR procedure (Algorithm 4) brings the potential numerical instability. Many elaborate and proven techniques for robust and efficient implementation of Krylov subspace techniques developed over the years could be considered for the second-order Krylov subspace $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$. The other subjects of further study include maintaining the orthogonality in the presence of finite precision arithmetic and a restarting strategy for solving the QEP by the SOAR method.

Krylov subspaces have an important characterization in terms of univariate matrix polynomials. Convergence theory of a Krylov subspace–based method has been established based on the theory of univariate polynomials and the distribution of eigenvalues of the underlying matrix. In section 2, we showed the connection between the second-order Krylov subspace $\mathcal{G}_n(\mathbf{A}, \mathbf{B}; \mathbf{u})$ and the bivariate polynomials $p_j(\alpha, \beta)$. It is unclear whether it can be used to develop a convergence theory which is directly based on the distribution of the matrices $\mathbf{A}$ and $\mathbf{B}$.

A closely related problem to the central theme of this paper is that of model-order reduction of a second-order dynamical system. The problem is about how to produce a reduced-order system of the same second-order form. One pioneering work is due to Su and Craig [22] back to 1991. In recent years, this approach has been repeatedly applied, studied, and improved; for example, see [2, 14, 19, 20]. In particular, the dissertation work of Slone [19] has essentially extended Su and Craig's approach to the model reduction of high-order dynamical systems but is based the popular AWE (asymptotic waveform evaluation) approach as widely known in interconnect analysis of integrated circuits and computational electromagnetics. In a forthcoming work, we will examine the application of the SOAR method for the model reduction of a second-order dynamical system and its connections to those previous works.

Acknowledgments. We are grateful to Gerard Sleijpen, Tom Kowalski, and Leonard Hoffnung for providing the matrix data used in Examples 4, 5, and 6, respectively. We would like to thank Ming Gu, Beresford Parlett, and Rodney Slone for helpful discussions during the course of this work. We thank the referees for valuable comments and suggestions to improve the presentation of the paper. One of the referees provided us the reference [13].

REFERENCES

[1] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[2] Z. Bai, *Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems*, Appl. Numer. Math., 43 (2002), pp. 9–44.

[3] 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*, SIAM, Philadelphia, 2000.

[4] A. Bermúdez, R. G. Durán, R. Rodríguez, and J. Solomin, *Finite element analysis of a quadratic eigenvalue problem arising in dissipative acoustics*, SIAM J. Numer. Anal., 38 (2000), pp. 267–291.

[5] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
[6] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[7] L. Hoffnung, R. C. Li, and Q. Ye, *Krylov type subspace methods for matrix polynomials*, Linear Algebra Appl., to appear.
[8] U. B. Holz, G. Golub, and K. H. Law, *A Subspace Approximation Method for the Quadratic Eigenvalue Problem*, Technical report SCCM-03-01, Stanford University, Stanford, CA, 2003.
[9] T. Kowalski, *Extracting a Few Eigenpairs of Symmetric Indefinite Matrix Pencils*, Ph.D. thesis, University of Kentucky, Lexington, KY, 2000.
[10] P. Lancaster, *Lambda-Matrices and Vibrating Systems*, Pergamon Press, Oxford, UK, 1966.
[11] 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.
[12] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice–Hall, Englewood Cliffs, NJ, 1980; revised reprint, Classics Appl. Math. 20, SIAM, Philadelphia, 1997.
[13] F. A. Raeven, *A new Arnoldi approach for polynomial eigenproblems*, in Proceedings of the Copper Mountain Conference on Iterative Methods, http://www.mgnet.org/mgnet/Conferences/CMCIM96/Psfiles/raeven.ps.gz, 1996.
[14] D. Ramaswamy and J. White, *Automatic generation of small-signal dynamic macromodels from 3-D simulation*, in Technical Proceedings of the Fourth International Conference on Modeling and Simulation of Microsystems, Nano Science and Technology Institute (NSTI), 2000, pp. 27–30.
[15] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Halsted Press, New York, 1992.
[16] Y. Saad, *Iterative Methods for Linear Systems*, PWS, Boston, 1996.
[17] 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.
[18] G. L. G. Sleijpen, H. A. van der Vorst, and M. B. van Gijzen, *Quadratic eigenproblems are no problem*, SIAM News, 29 (1996), pp. 8–9.
[19] R. D. Slone, *Fast Frequency Sweep Model Order Reduction of Polynomial Matrix Equations Resulting from Finite Element Discretization*, Ph.D. thesis, Ohio State University, Columbus, OH, 2002.
[20] R. D. Slone, R. Lee, and J.-F. Lee, *Broadband model order reduction of polynomial matrix equations using single-point well-conditioned asymptotic waveform evaluation: Derivations and theory*, Internat. J. Numer. Methods Engrg., 58 (2003), pp. 2325–2342.
[21] G. W. Stewart, *Matrix Algorithms, Volume* II: *Eigensystems*, SIAM, Philadelphia, 2001.
[22] T.-J. Su and R. R. Craig, Jr., *Model reduction and control of flexible structures using Krylov vectors*, J. Guidance Control Dynam., 14 (1991), pp. 260–267.
[23] F. Tisseur, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra Appl., 309 (2000), pp. 339–361.
[24] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 3, pp. 660–678

INEXACT MATRIX-VECTOR PRODUCTS IN KRYLOV METHODS FOR SOLVING LINEAR SYSTEMS: A RELAXATION STRATEGY*

AMINA BOURAS† AND VALÉRIE FRAYSSÉ‡

Abstract. Embedded iterative linear solvers are being used more and more often in linear algebra. An important issue is how to tune the level of accuracy of the inner solver to guarantee the convergence of the outer solver at the best global cost. As a first step towards the challenging goal of controlling embedded linear solvers, inexact Krylov methods are used as a model of inner-outer iterations with external Krylov scheme. This paper experimentally shows that Krylov methods for solving linear systems can still perform very well in the presence of carefully monitored inexact matrix-vector products. This surprising behavior of inexact Krylov methods, as opposed to Newton-like methods, is investigated in detail, and potentially important applications are mentioned. A new relaxation strategy for the inner accuracy is proposed for Krylov methods with inexact matrix-vector products; its efficiency is supported by a wide range of numerical experiments on different algorithms and contrasted against other potential approaches.

Key words. inner-outer iterations, Krylov method, inexact matrix-vector products, embedded iterative linear solvers.

AMS subject classifications. 65F10, 65F15, 15A06, 15A18

DOI. 10.1137/S0895479801384743

1. About inner-outer iterations in linear algebra.

Iterative processes are widely used in linear algebra for treating large sets of data. It is becoming more and more common that one iterative solver has to be embedded in an outer one: this is the case, for instance, for solving eigenproblems with inverse iterations or with a Krylov method with invert. Each outer step (that is, each step of the eigensolver) requires the solution of a linear system which, if too large, must be solved in turn with an iterative method (inner steps). The question arises then: *What is the best strategy for stopping the inner iterations in order to ensure the convergence of the outer iterations while minimizing the global computational cost?* This question has been partially addressed by numerical experts since the eighties in the context of Newton-like and, more generally, fixed point methods [5, 7, 11]. It is generally concluded, as one could expect, that the accuracy of the inner iteration is a threshold for the convergence of the outer process: it cannot be weakened when the outer process comes closer to the solution. The proposed strategies for monitoring the inner iterations have so far been very problem- and method-dependent. More recently in the late nineties, the different behavior of embedded solvers involving a Krylov outer process has been emphasized [8, 12, 13, 19]. For instance, the strikingly different behaviors of inverse iterations and Krylov methods for the solution of eigenproblems with respect to inner iterations are mentioned in [13] for symmetric and in [3] for nonsymmetric matrices. It is observed, as for Newton-like methods, that inverse iterations require inner iterations

*Received by the editors January 25, 2001; accepted for publication (in revised form) by G. H. Golub June 27, 2004; published electronically March 3, 2005.
http://www.siam.org/journals/simax/26-3/38474.html

†Université Toulouse I and CERFACS, 42 av. Gaspard Coriolis, 31057 Toulouse cedex 1, France (bouras@cerfacs.fr). The work of this author was supported by CNES (Centre National d'Etudes Spatiales).

‡Kvasar Technology LLC., 372 Marlborough Street, Apt. 10, Boston, MA 02115 (valerie@spydre.com). This work was performed while the author was a researcher at CERFACS, 42 av. Gaspard Coriolis, 31057 Toulouse cedex 1, France.

to be more and more accurate while approaching the solution. But for Lanczos or Arnoldi methods, on the contrary, the first Krylov vectors need to be known with full accuracy, and this accuracy can be *relaxed* as the convergence proceeds. A strategy for monitoring the accuracy of inner iterations is proposed in the framework of symmetric eigenvalue problems with homogeneous linear constraints in [13].

In order to examine closely the seemingly counterintuitive behavior of inner-outer Krylov methods, we investigate in this paper the behavior of Krylov methods when information on the matrix may be partially unavailable, resulting in the use of inexact basis vectors. In order to focus on the root phenomenon, we set up this study in the context of linear systems (see [3] for a similar work on eigenproblems). We will show that, when the inaccuracy of the basis vectors is controlled by a carefully designed relaxation strategy, the Krylov method for solving linear systems can still perform with a remarkable efficiency. It is beyond the scope of this paper to provide a detailed comparison of the many possible relaxation strategies, although we will contrast a few of them in order to expose the reasons that motivated our choices. The inexact Krylov scheme serves here as a simple model for understanding more complex embedded iterative schemes where the outer iteration would be a Krylov method.

1.1. Some important applications. Understanding the effects of inexact matrix-vector products can have many applications. As such, inexact matrix-vector products are encountered in multipole methods, which have recently become popular in the numerical solution of large electromagnetism problems. The main feature of multipole methods is that the matrix-vector product is computed through an expansion whose order can be monitored [9]. In such a situation, the matrix is not formed explicitly and its application to a vector is computed within some level of accuracy only. The higher the order, the more expensive the product. Therefore, relaxation on the accuracy of the matrix-vector products would directly decrease the cost of the iterative method.

Moreover, embedded iterations involving an outer Krylov solver also fit within the scope of this study. This is the case, for instance, of the Arnoldi method with invert for computing the smallest eigenvalue of a large sparse matrix. Although the matrix A may be known exactly, the matrix A^{-1} is not: in order to compute an orthonormal basis for the Krylov space

$$\text{span}\{v_1, A^{-1}v_1, A^{-2}v_1, \dots\}$$

one needs to solve a linear system $Az = v_k$ in order to get the next Krylov vector v_{k+1}. If one uses an approximate linear solver (such as an iterative method), the approximate solution $\hat{z}$ satisfies $(A + \Delta A_k)\hat{z} = v_k$. The backward error analysis viewpoint amounts to considering that the algorithm is applied to an approximation $(A + \Delta A_k)$ of A changing at each step. Again, it is possible to relax the accuracy on $\hat{z}$ as long as the outer process converges so that both the cost of the inner iterations and the global cost are reduced [3].

Another application of importance arises in the context of domain decomposition methods for partial differential equations (PDEs). For large problems, the local subproblems induced by the decomposition have to be solved by an iterative process which is embedded in the outer iterative process used to solve the Schur complement equation. The results from the present work readily apply. In [4], it is shown that, when the Schur complement equation is solved by the conjugate gradient (CG) method, a significant reduction of the computational cost can be obtained from a relaxation strategy on the inner iteration accuracy.

1.2. Outline. This paper is organized as follows. Section 2 defines the basic inexact Krylov scheme derived from the GMRES algorithm. A relaxation strategy, chosen for its good performance, is then described before its numerical behavior is illustrated in section 3 on a set of test matrices taken from the Harwell–Boeing collection and on various algorithms including CG (for short-recurrence algorithms), GMRES, and BiCGStab.

Then in section 4, we give the reasons that lead to the choice of the proposed relaxation strategy which we contrast against other possible relaxation schemes. We also discuss potential scaling issues and give some considerations on a practical implementations of such a strategy, based on our experience of real-world applications. The last section concludes this work by giving some hints and tracks to be investigated in order to progress towards a fully justified approach of inexact Krylov schemes.

2. Inner-outer iterations in Krylov methods.

2.1. The basic inexact Krylov scheme. We consider the GMRES method for solving the linear system $Ax = b$, where $A \in \mathbb{C}^{n\times n}$ and x and b are two vectors of $\mathbb{C}^n$. This method, detailed in Algorithm 1, is one of the simplest and, at the same time, one of the most widely used Krylov-type methods for solving a linear system [17].

Algorithm 1. GMRES.

$r_0 = b - Ax_0$; $\beta = \|r_0\|_2$
$v_1 = r_0/\beta$
for $k = 1, 2, \ldots,$ **do**
  $z = Av_k$
  for $i = 1$ `to` k **do**
    $h_{ik} = v_i^* z$
    $z = z - h_{ik} v_i$
  end for
  $h_{k+1k} = \|z\|$
  $v_{k+1} = z/h_{k+1k}$
  Solve the least-squares problem $\min \left\|\beta e_1 - \bar{H}_k y\right\|_2$ for y
  Set $x_k = x_0 + V_k y$
  Exit if satisfied
end for

Let x_0 be the initial guess and let $r_0 = b - Ax_0$ be the initial residual. We denote by e_k the kth canonical vector and by $\|\cdot\|$ the Euclidean norm. The GMRES method builds a basis $V_k = [v_1, \ldots, v_k]$ for the Krylov space

$$\mathcal{K}_k = \text{span}\{r_0, Ar_0, \ldots, A^{k-1} r_0\},$$

and the Hessenberg matrix $H_k = V_k^* A V_k \in \mathbb{C}^{k\times k}$ is the orthogonal projection of A onto $\mathcal{K}_k$. Let $\bar{H}_k \in \mathbb{C}^{(k+1)\times k}$ be the H_k matrix augmented by the row vector $h_{k+1k} e_k^T$. The Krylov process can be viewed as the QR decomposition

$$[v_1 \; AV_k] = V_{k+1}[e_1 \; \bar{H}_k].$$

The outer iteration corresponds to the addition of a new Krylov vector in the basis $z = Av_k$. As such, the GMRES method does not show any inner iteration. To simulate the effects of an inner iteration, we perform inexact matrix-vector products

Algorithm 2. GMRES with inexact matrix-vector products.

Set the initial guess $x_0 = 0$
$r_0 = b - Ax_0 = b$; $\beta = \|r_0\|_2$
$\hat{v}_1 = r_0 / \|r_0\|$;
for $k = 1, 2, \ldots,$ **do**
  $\hat{z} = (A + \Delta A_k)\hat{v}_k$
  for $i = 1$ `to` k **do**
    $\hat{h}_{ik} = \hat{v}_i^* \hat{z}$
    $\hat{z} = \hat{z} - \hat{h}_{ik}\hat{v}_i$
  end for
  $\hat{h}_{k+1k} = \|\hat{z}\|$
  $\hat{v}_{k+1} = \hat{z}/\hat{h}_{k+1k}$
  Solve the least-squares problem $\min \|\beta e_1 - \overline{\widehat{H}}_k y\|$ for y
  Set $x_k = x_0 + \widehat{V}_k y$ and $r_k = b - Ax_k$
  Exit if satisfied
end for

in the computation of the Krylov vectors, as shown in Algorithm 2. More precisely, the vector $\hat{v}_{k+1}$ is obtained by computing

$$\hat{z} = (A + \Delta A_k)\hat{v}_k$$

and then orthonormalizing $\hat{z}$ against all the previous vectors $\hat{v}_i$, $i = 1, \ldots, k$. Although one could think of perturbing directly the result of the matrix-vector product Av_k, we think it is more realistic to apply the perturbation on the matrix A, thus modeling the effect of incomplete information available from the matrix itself (as would be the case in the multipole application, for instance) in addition to setting our analysis in a natural backward analysis framework. The matrix ΔA_k is a perturbation matrix satisfying some prescribed properties. A similar trick was used in [12] to simulate an inexact preconditioner for CG. Therefore, instead of working on the Krylov space $\mathcal{K}_k$, we use instead the space

$$\widehat{\mathcal{K}}_k = \text{span}\{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_k\},$$

where $\hat{z}$ is orthonormalized against $\widehat{V}_k = [\hat{v}_1, \ldots, \hat{v}_k]$ to produce $\widehat{V}_{k+1}$. We set $\hat{v}_1 = v_1 = r_0/\|r_0\|$. The underlying QR decomposition is turned into

$$[\hat{v}_1 \; A\widehat{V}_k] + [0 \;\; \Delta A_1\hat{v}_1, \ldots, \Delta A_k\hat{v}_k] = \widehat{V}_{k+1}[e_1 \; \overline{\widehat{H}}_k \,].$$

Therefore, the Hessenberg matrix $\overline{\widehat{H}}_k$ does not represent anymore the projection of A onto $\widehat{\mathcal{K}}_k$.

The aim of this work is to propose and experiment with a strategy which monitors the perturbations ΔA_i (in structure and in size) in such a way that the outer process still converges within only a few extra iterations (at most).

A preliminary remark is that if all the ΔA_i are equal to the same matrix E, then one solves in fact the linear system $(A+E)x = b$. The backward error for the computed solution $\tilde{x}$ with respect to the original system $Ax = b$ is $\|E\tilde{x}\|_2 / (\|A\|_2 \|\tilde{x}\|_2)$. It is bounded above by $\|E\|_2 / \|A\|_2$ and should not be much smaller unless $\tilde{x}$ specifically lies in the subspace associated with the smallest singular values of E. Similarly,

numerical experiments show that if all the ΔA_i differ but stay equal in norm to $\eta \|A\|$, then in most of the cases the computed solution of the linear system is computed with a backward error of the order of η. This is not unexpected if one thinks of the GMRES process applied in finite precision: each matrix-vector product is indeed computed within a limited accuracy. It amounts to using a slightly perturbed matrix at each step, where the relative perturbation size is of the order of machine precision.

More surprisingly, this paper will show the remarkable fact that it is indeed possible to let the size of the perturbations ΔA_k grow significantly throughout the outer process. This fact is supported by a wide set of numerical experiments. An important feature of our approach is that it is set in the framework of the backward error analysis. We have chosen indeed to express the inexact matrix-vector product under the form $\hat{z} = (A + \Delta A_k)\hat{v}_k$. This is important for two reasons: first, because the backward error analysis is the tool of choice for understanding computational processes with inexact data; second, because this powerful framework naturally applies to inner-outer processes. With this modelization, we are then able to treat in a unified way inexact matrix-vector products (such as in the multipole methods for electromagnetism), or inner-outer methods with outer Krylov scheme (where the inaccuracy of the inner scheme can be interpreted if not monitored in terms of a backward error on the matrix).

In addition, we will be dealing with perturbations having a relative size always larger than machine precision, and often significantly larger: the observed phenomena are primarily due to the perturbations we apply and are not artifacts due to the finite precision of the computer arithmetic.

2.2. A relaxation strategy on the inner accuracy. Let us now define a strategy to increasingly perturb the matrix-vector product as long as the outer process converges. Let r_k be the residual $Ax_k - b$ at step k. Let η be the final tolerance required for the solution of the linear system. More precisely we aim at computing a solution $\tilde{x}$ with a backward error $\|A\tilde{x} - b\| / (\|A\| \, \|\tilde{x}\|)$ smaller than η.

The proposed strategy for performing the inexact matrix-vector products is the following. Let α_k be the scalar defined by

$$\alpha_k = \frac{1}{\min(\|r_{k-1}\|, 1)}.$$

Each matrix-vector product involved in the computation of the Krylov basis is replaced by

$$\hat{z} = (A + \Delta A_k)v_k,$$

where ΔA_k is a random matrix satisfying

$$\|\Delta A_k\| = \varepsilon_k \|A\|, \quad \varepsilon_k = \min(\alpha_k \eta, 1). \tag{2.1}$$

Therefore at each step the applied perturbation is always larger than or equal to the targeted tolerance η, and always smaller than or equal to 1, in a normwise relative sense:

$$\frac{\|\Delta A_k\|}{\|A\|} \in [\eta, 1]. \tag{2.2}$$

We have forced $\varepsilon_k \leq 1$ to avoid too large relative perturbations that would not retain information on A. We see that when r_k decreases, ε_k increases (or stays at 1). The

first vectors of the Krylov basis are computed with a backward error of the order of the targeted tolerance η as long as the norm of the residual is larger than 1. On the contrary, the last vectors may correspond to relatively large perturbations of A. The accuracy of the matrix-vector products is therefore *relaxed* while the outer process converges.

It has to be noted that α_k is an absolute quantity because it involves the residual without normalization, whereas $\varepsilon_k = \|\Delta A_k\| / \|A\|$ is relative. As discussed in section 4, we have found this choice to be the best in our experimental practice.

We may impose additionally some structure on ΔA_k. We have performed tests with dense matrices ΔA_k and with matrices having the same sparsity pattern of A. Both approaches gave similar results. We report only the results obtained when preserving the sparsity structure (`sprand` from MATLAB).

3. Numerical experiments.

3.1. Test algorithms. The first algorithm we present is the GMRES method; its inexact version is described above in Algorithm 2. However, since the full GMRES does not always converge on our set of test matrices within a reasonably small projection size, we also use its restarted version, denoted GMRES(m). Let m be the value of the restart parameter, that is, the maximal size allowed for the projection. The restarted method with inexact matrix-vector products is detailed in Algorithm 3. The strategy for choosing ΔA_k for $k > 0$ is the same as for the full GMRES. Let $\Delta A_k^{(j)}$ be the perturbation introduced at step k of the jth restart; then

$$\|\Delta A_k^{(j)}\| = \varepsilon_k^{(j)} \|A\| \quad \text{with} \quad \varepsilon_k^{(j)} = \min(\alpha_k^{(j)}\eta, 1)$$

with

$$\begin{cases} \alpha_1^{(j)} = \dfrac{1}{\min(\|r_m^{(j-1)}\|, 1)}, \ \alpha_1^{(1)} = 1, \\ \alpha_k^{(j)} = \dfrac{1}{\min(\|r_{k-1}^{(j)}\|, 1)} \quad \text{if } k > 1. \end{cases}$$

Therefore, the accuracy of $\hat{z}_1$ in Algorithm 3 is controlled by the reciprocal of the residual associated with the solution obtained at the end of the previous restart $(j-1)$.

However, the residual r_0 which initiates each restart is also computed inexactly at the targeted tolerance: $r_0 = b - (A + \Delta A_0)x_0$ with $\|\Delta A_0\| = \eta \|A\|$, as this is the only quantity in the algorithm that carries information about the right-hand side. In a context where the matrix A is accessed only via inexact matrix-vector products, it is important to be able to deal with an inexact r_0. But we observed that we could not allow perturbations of size larger than η on the computation of r_0.

However, this approach is quite different from the one chosen in [13], where the inner accuracy is increased at each restart while the projection size decreases.

To broaden our choice of algorithms, we have also implemented an inexact version of the following:

- CG, as a representative of short-term recurrence algorithms [14]. We would like to mention that we have also been able to combine successfully inexact CG with the inexact preconditioner proposed by Golub and Ye [12], which is an additional illustration of the remarkable robustness of Krylov methods.
- BiCGStab. We apply the same perturbation strategy as for GMRES. However, since one iteration of BiCGStab involves two matrix-vector products, we have chosen to use the same perturbation matrix for both products.

Algorithm 3. GMRES(m) with inexact matrix-vector products.

Set the initial guess $x_0 = 0$
for $j = 1, 2, \ldots,$ **do** {The subscript (j) is omitted}
  $r_0 = b - (A + \Delta A_0)x_0$; $\beta = \|r_0\|_2$
  $\hat{v}_1 = r_0 / \|r_0\|$
  for $k = 1, 2, \ldots, m$ **do**
    $\hat{z_k} = (A + \Delta A_k)\hat{v}_k$
    for $i = 1$ `to` k **do**
      $\hat{h}_{ik} = \hat{v}_i^* \hat{z_k}$
      $\hat{z_k} = \hat{z_k} - \hat{h}_{ik}\hat{v}_i$
    end for
    $\hat{h}_{k+1k} = \|\hat{z_k}\|$
    $\hat{v}_{k+1} = \hat{z_k}/\hat{h}_{k+1k}$
    Solve the least-squares problem $\min \|\beta e_1 - \widehat{H}_k(1 : k+1, 1 : k)y\|$ for y
    Set $x_k = x_0 + \widehat{V}_k y$ and $r_k = b - Ax_k$
    Exit if satisfied
  end for
  Set $x_0 = x_m$
end for

Finally, we may also need a preconditioner that we usually take as an incomplete LU factorization with some threshold t [16]. We denote it by ILU(t). The preconditioner is applied on the left after the inexact matrix-vector product.

We present a series of experiments done with MATLAB 5 on a set of matrices taken from the Harwell–Boeing collection [6]. The right-hand side has been computed so that the exact solution is the vector of all ones.

3.2. Convergence process under inexact matrix-vector products. A typical observed behavior is shown in Figure 3.1 for the matrix `e05r0400` of order 236. In this case we use GMRES(m) with a restart $m = 10$ and ILU(10^{-3}). The iteration number is shown on the horizontal axis and should be read in the following sense: iteration 25 ($= (3 - 1) \times m + 5$) means the fifth step of the third restart. Each figure also bears the condition number and the norm of the matrix. The line with "$\circ$" is the convergence curve with exact matrix-vector products, and the line with "$+$" corresponds to inexact products. By convergence curve we mean the evolution of the normwise backward error associated with the current estimate of the solution. The line with "$\times$" represents the relative size $\varepsilon_k = \|\Delta A_k\| / \|A\|$ of the perturbation imposed on the matrix-vector product at each outer iteration. Finally the straight horizontal line represents the final targeted tolerance η.

We see in Figure 3.1 that the first 7 vectors of the first restart have been computed with a perturbation size equal to η: this is because the norm of the outer residual is still ≥ 1. As soon as the norm of the residual becomes less than 1, then $\varepsilon_k = \eta / \|r_k\| > \eta$ and the matrix-vector product is computed with less and less accuracy, as shown by the increasing line ($\times$). In this case, we see that the convergence curves with exact ($\circ$) and inexact ($+$) products cannot be distinguished (at the graphical level) before the targeted tolerance is reached. This amazing fact is observed in many experiments. When the backward error associated with the current iterate becomes of the order of η, the convergence curve corresponding to inexact products stalls at a value of the order of η, as expected. In this particular example, the linear system is solved

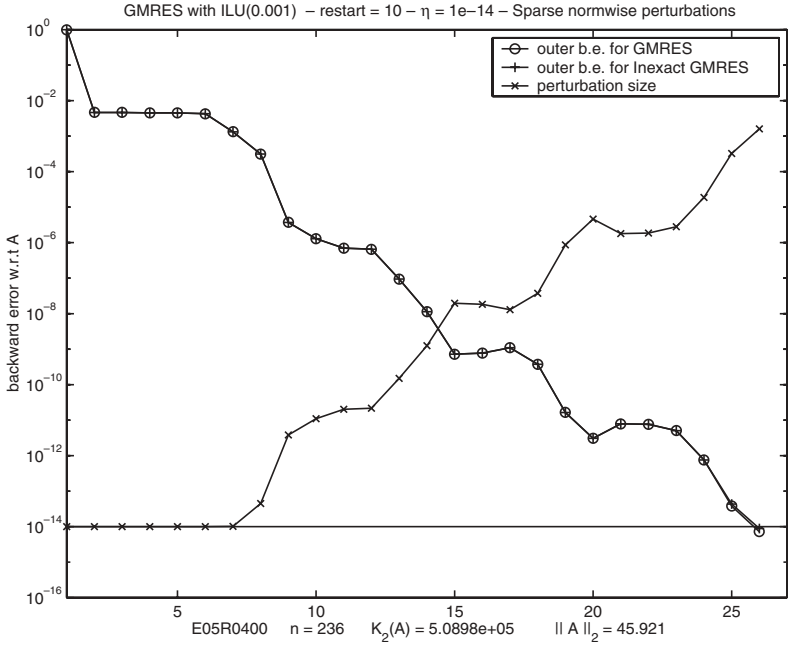


FIG. 3.1. *Exact* (∘) *vs. inexact* (+) *matrix-vector products in GMRES*(m). $\eta = 10^{-14}$, $m = 10$.

in 25 steps (to reach a tolerance of 10^{-14}), whether the matrix-vector products are exact or perturbed. In practice, perturbing the matrix-vector products should result in an increase in the number of steps. It is remarkable to see that many (19 out of 26 in this example) of the Krylov vectors can be significantly perturbed (up to 10^{-3} in this example, to be compared to $\eta = 10^{-14}$) without altering the convergence process. Moreover, it was quite unexpected to see that all the Krylov vectors of the last restarts are computed with a high perturbation apart from the first vector $v_1 = r_0/\|r_0\|$, which is perturbed at the level of η.

3.3. Summary of the experiments. Let us now browse through a variety of matrices while testing several Krylov solvers. We first need to state more precisely the definition of convergence. Indeed, it may seem quite ambitious to expect the backward error to be of the order of η when each matrix-vector product is perturbed by at least the order of η. It is somehow like requiring a backward error to be less than machine precision in finite precision arithmetic. Therefore it is already very satisfactory to reach a final backward error of the order of 10η or even $10^2\eta$. To illustrate this point, we record the number of iterations N_1 (resp., N_{10} and N_{100}) necessary for the backward error to become smaller than η (resp., 10η and 100η) when possible. These numbers have to be compared with the number of iterations N_{ex} for the backward error to become smaller than η with exact products, to serve as a reference.

Table 3.1 (resp., Table 3.2) summarizes the experiments performed with GMRES (resp., GMRES(m)) according to Algorithm 2 (resp., Algorithm 3). Using a preconditioned restarted version of GMRES allows us to solve a wider range of linear systems (with or without a relaxation strategy), which is why Table 3.2 contains more test matrices than Table 3.1. Tables 3.3 and 3.4 are devoted to the results obtained with similar experiments on the CG and the BiCGStab methods, respectively. When an

TABLE 3.1
GMRES with inexact matrix-vector products.

				Inexact products			
Matrix	n	η	N_{ex}	N_1	N_{10}	N_{100}	Figure
ARC130	130	10^{-14}	16	16	15	14	
	130	10^{-11}	12	12	5	5	
FS_183_6	183	10^{-12}	40	44	32	23	
	183	10^{-14}	44	47	44	42	
GRE115	115	10^{-14}	80	–	77	75	
	115	10^{-05}	51	–	46	30	
GRE185	185	10^{-12}	161	–	159	158	
	185	10^{-10}	158	–	156	154	Fig. A.1
WEST0132	132	10^{-10}	130	130	124	114	
	132	10^{-08}	114	111	91	16	

TABLE 3.2
GMRES(m) with inexact matrix-vector products.

						Inexact products			
Matrix	n	t	m	η	N_{ex}	N_1	N_{10}	N_{100}	Figure
e05r0400	236	10^{-3}	10	10^{-14}	26	26	25	24	Fig. 3.1
e05r0000	236	10^{-2}	20	10^{-14}	95	110	93	76	
	236	10^{-2}	20	10^{-10}	59	69	57	55	
	236	10^{-2}	20	10^{-06}	36	63	34	30	
GRE115	115	10^{-1}	10	10^{-10}	18	18	17	15	
GRE185	185	10^{-2}	10	10^{-14}	91	–	80	73	
	185	10^{-2}	10	10^{-10}	59	166	53	43	
	185	10^{-2}	15	10^{-10}	29	155	39	27	Fig. A.3
GRE343	343	10^{-1}	10	10^{-10}	42	43	38	33	
CAVITY03	317	10^{-3}	10	10^{-10}	24	24	21	16	
PDE225	225	10^{-1}	10	10^{-14}	26	27	24	22	
	225	10^{-1}	10	10^{-13}	24	24	22	21	
	225	10^{-1}	10	10^{-10}	19	20	18	16	
SAYLR1	238	10^{-1}	10	10^{-13}	131	131	110	90	Fig. A.2
	238	10^{-1}	10	10^{-10}	81	91	66	51	
UTM300	300	10^{-3}	15	10^{-11}	56	–	–	46	
	300	10^{-3}	15	10^{-06}	30	–	28	16	
	300	10^{-3}	20	10^{-11}	34	–	28	21	
	300	10^{-3}	20	10^{-06}	18	–	17	16	
WEST0381	381	10^{-2}	10	10^{-10}	29	30	28	24	
	381	10^{-2}	10	10^{-06}	17	16	15	11	
BFW398A	398	10^{-1}	20	10^{-12}	148	–	138	116	
	398	10^{-1}	20	10^{-08}	93	–	73	62	

incomplete LU preconditioner with threshold is applied to the left of the system, the value t of the threshold is also reported in the tables. In the appendix, we present four figures of the same kind as Figure 3.1 selected from the experiments among those reported in the three tables. The interested reader is referred to [2] for the complete set of plots associated with Tables 3.1, 3.2, and 3.4.

In the experiments on GMRES and GMRES(m) with inexact products, we have always been able to obtain a backward error at least smaller than 100η with GMRES and GMRES(m). Even more, the cases where the backward error could not be lower than 10η are very seldom. It is also very interesting that the convergence with inexact products is achieved within a number of iterations which is of the order of the one obtained with exact products. Exceptionally, it may even happen that

TABLE 3.3
CG with inexact matrix-vector products.

					Inexact products			
Matrix	n	t	η	N_{ex}	N_1	N_{10}	N_{100}	Figure
BCSSTK27	1224	10^{-2}	10^{-12}	50	52	48	45	
	1224	10^{-2}	10^{-14}	55	57	53	51	
BCSSTK14	1806	5.10^{-3}	10^{-12}	54	–	51	47	
	1806	5.10^{-3}	10^{-14}	60	–	58	55	
	1806	10^{-2}	10^{-12}	69	–	66	61	
BCSSTK15	3948	5.10^{-3}	10^{-12}	145	–	141	131	
	3948	10^{-1}	10^{-12}	221	–	224	210	
S1RMQ4M1	5489	10^{-2}	10^{-12}	135	147	129	116	
	5489	5.10^{-2}	10^{-12}	245	284	256	236	Fig. A.5
	5489	10^{-1}	10^{-08}	210	224	193	158	
	5489	10^{-1}	10^{-10}	246	260	232	213	
	5489	10^{-1}	10^{-12}	283	296	267	248	

TABLE 3.4
BiCGStab with inexact matrix-vector products.

					Inexact products			
Matrix	n	t	η	N_{ex}	N_1	N_{10}	N_{100}	Figure
BFW398A	236	10^{-1}	10^{-12}	84	–	–	–	Fig. A.4
	236	10^{-3}	10^{-12}	11	–	10	10	
CAVITY03	317	10^{-3}	10^{-10}	23	–	–	18	
	317	10^{-3}	10^{-08}	16	–	–	13	
e05r0000	236	10^{-2}	10^{-10}	51	–	–	43	
	236	10^{-2}	10^{-06}	40	–	34	26	
e05r0400	236	10^{-3}	10^{-12}	20	–	–	17	
	236	10^{-3}	10^{-06}	10	–	10	2	
GRE115	115	10^{-1}	10^{-12}	24	27	24	24	
	115	10^{-1}	10^{-09}	21	–	20	18	
GRE185	185	10^{-2}	10^{-10}	34	–	–	–	
GRE343	343	10^{-1}	10^{-10}	38	–	36	32	
PDE225	225	10^{-1}	10^{-13}	25	25	22	21	
	225	10^{-1}	10^{-10}	20	20	18	16	
SAYLR1	238	10^{-1}	10^{-13}	44	–	44	39	
	238	10^{-1}	10^{-10}	32	43	37	36	

GMRES or GMRES(m) with perturbed matrix-vector products converge faster than their exact counterparts by a few iterations (see WEST0132 in Table 3.1 and WEST0381 in Table 3.2). In some cases, the convergence of the perturbed algorithm is achieved with many extra iterations: see, for instance, GRE185 in Table 3.2. But usually in those cases convergence within 10η is always achieved with a number of iterations comparable to that for the exact algorithm. Therefore, the overhead in terms of iterations induced by the inexact matrix-vector products is quite low. This is all the more remarkable because the size of the perturbations allowed by the strategy described in (2.1) can grow fast and reach large values (see, for instance, Figure 3.1 or A.3). This shows that the Krylov process is robust to perturbations of the matrix-vector products provided that the first Krylov vectors are computed with the full targeted accuracy.

The results obtained for CG with inexact matrix-vector products (symmetry is not maintained) in Table 3.3 confirm that the observed robustness is inherent to Krylov processes and should be shared by other numerical Krylov schemes. The results,

obtained on matrices of larger size than in the previous cases, show that decreasing the backward error below 10η was always achievable, and that the threshold of η was reached in more than half of the cases with a reasonable overhead in terms of additional steps. We refer the interested reader to [4] for a more detailed evaluation of the gain obtained with relaxation schemes for CG in the context of domain decomposition methods. The picture is slightly less clear with BiCGStab; if in most of the cases the backward error is smaller than 100η, we have also encountered a few examples where the backward error could not decrease significantly. Anyway, the possibility of applying a relaxation strategy still holds: the gain is just not as high.

3.4. Practical implementation. It has to be noted that the definition of ε_k in (2.1) relies upon information about the true residual $r_{k-1} = b - Ax_{k-1}$ at step $k-1$. However, in the context of avoiding exact matrix-vector products, $\|r_{k-1}\|$ needs to be replaced by a quantity directly available from the algorithm, such as the GMRES residual, which is a by-product of the QR factorization of the augmented Hessenberg matrix arising in the least-squares solution. However, whether this GMRES residual still gives information about the true residual when the Krylov space is perturbed has to be tested carefully. Experiments with embedded CG algorithms in domain decomposition techniques performed on realistic PDE problems are encouraging: a comparison of the true and the by-product residuals during the relaxation strategy can be found in [4] for CG, where it appears that the difference between both quantities is not particularly affected by the introduction of the relaxation scheme.

Note also that the quantities we have reported as the "true" or "exact" residual or any by-product quantity given by the algorithm were indeed computed in finite precision. However, we are dealing here with matrix perturbations of relatively large size (see (2.2)) so that effects of finite precision should not be dominant.

4. Variations on relaxation strategies.

4.1. Other relaxation schemes. The relaxation strategy proposed above basically varies as the reciprocal of the residual. One can legitimately wonder why, and whether other indexations such as those on the reciprocal of the square of the residual or its square root, for instance, would not be equally applicable.

A strategy indexed on the reciprocal of the square of the residual would generate larger inaccuracies, and our practice has shown us that not enough information would be retained to ensure the global convergence of the outer Krylov scheme in most of our attempts.

On the contrary, it is expected that a relaxation strategy based on the reciprocal of the square root of the residual would work in a larger number of cases than the strategy proposed above since the size of the perturbation would be smaller for a similar convergence pattern. This is observed on Tables 4.1 and 4.2, which offer the same test cases as in Tables 3.1 and 3.2 but with a relaxation scheme where α_k has been replaced by

$$\gamma_k = \frac{1}{\min(\sqrt{\|r_k\|}, 1)}.$$

It is clear that this more conservative strategy recovers the global convergence of GMRES (see `GRE115` and `GRE185`) or GMRES(m) (see `UTM300` and `BFW398A`) in most of the cases where the strategy indexed on the reciprocal of the residual would fail to ensure a final backward error smaller than η or would require a very high overhead

TABLE 4.1
GMRES with inexact matrix-vector products. Strategy indexed on $1/\sqrt{\|r_k\|}$.

				Inexact products		
Matrix	n	η	N_{ex}	N_1	N_{10}	N_{100}
ARC130	130	10^{-14}	16	16	15	14
	130	10^{-11}	12	12	5	5
FS_183_6	183	10^{-12}	40	44	32	23
	183	10^{-14}	44	47	43	42
GRE115	115	10^{-14}	80	80	77	75
	115	10^{-05}	51	51	46	30
GRE185	185	10^{-12}	161	161	159	158
	185	10^{-10}	158	158	156	154
WEST0132	132	10^{-10}	130	130	124	114
	132	10^{-08}	114	111	91	16

TABLE 4.2
GMRES(m) *with inexact matrix-vector products. Strategy indexed on* $1/\sqrt{\|r_k\|}$.

						Inexact products		
Matrix	n	t	m	η	N_{ex}	N_1	N_{10}	N_{100}
e05r0400	236	10^{-3}	10	10^{-14}	26	28	25	24
e05r0000	236	10^{-2}	20	10^{-14}	95	96	93	76
	236	10^{-2}	20	10^{-10}	59	72	57	55
	236	10^{-2}	20	10^{-06}	36	38	34	30
GRE115	115	10^{-1}	10	10^{-10}	18	18	17	15
GRE185	185	10^{-2}	10	10^{-14}	91	94	81	73
	185	10^{-2}	10	10^{-10}	59	61	54	43
	185	10^{-2}	15	10^{-10}	29	30	28	27
GRE343	343	10^{-1}	10	10^{-10}	42	43	38	33
CAVITY03	317	10^{-3}	10	10^{-10}	24	24	20	16
PDE225	225	10^{-1}	10	10^{-14}	26	27	24	22
	225	10^{-1}	10	10^{-13}	24	24	22	21
	225	10^{-1}	10	10^{-10}	19	20	18	16
SAYLR1	238	10^{-1}	10	10^{-13}	131	140	111	99
	238	10^{-1}	10	10^{-10}	81	91	66	51
UTM300	300	10^{-3}	15	10^{-10}	52	53	46	41
	300	10^{-3}	15	10^{-06}	30	–	61	19
	300	10^{-3}	20	10^{-11}	34	35	33	21
	300	10^{-3}	20	10^{-06}	18	–	17	17
WEST0381	381	10^{-2}	10	10^{-10}	29	29	28	26
	381	10^{-2}	10	10^{-06}	17	16	15	11
BFW398A	398	10^{-1}	20	10^{-12}	148	152	137	121
	398	10^{-1}	20	10^{-08}	93	82	62	62

of outer iterations to meet this criterion. In the cases where the strategy indexed on $1/\|r_k\|$ succeeds in ensuring the global convergence, the strategy with γ_k may require fewer outer iterations (see GMRES(m) on GRE185, for instance) but not always (see GMRES(m) on e05r0000). However, both strategies behave similarly when one considers obtaining a final backward error of the order of 10η as a satisfactory final goal. In order to achieve a final backward error of η on a given linear system, one may apply, for instance,

- a relaxation strategy indexed on the reciprocal of the square root of the residual (i.e., using γ_k) and η;
- a relaxation strategy indexed on the reciprocal of the residual and $\eta' = \eta/10$ with a stopping criterion chosen as $10\eta'$.

The quality of the computed solution would be the same, but the global computational cost may differ.

Indeed, many variants may be thought of as one starts playing with the parameters of the relaxation scheme, such as η, α_k, or the stopping criterion. It is particularly difficult to compare the different strategies on our model problem of inexact Krylov schemes because we do not have a good way to measure the performance of these strategies: the number of outer iterations itself is not a good criterion since it does not take into account the gain obtained from using inaccurate matrix-vector products. In practice, the choice of a good strategy will result from some trade-off between the cost of the inner iteration and the overhead on the outer iterations.

Therefore, so far we have been primarily interested in the achievable accuracy of the relaxation schemes. A general comparison of these schemes in terms of computational cost is beyond the scope of this paper. Such a study requires the knowledge of the inner process modeled here by a perturbation. We refer, for instance, to our work on domain decomposition methods with Giraud where the gain can be effectively measured in terms of a significant reduction of matrix-vector products [4].

The strategy which indexes the perturbation size on the reciprocal of the residual was therefore privileged because it is the one that allows the larger perturbations sizes while preserving the global convergence. If the inaccuracies allowed on the matrix-vector product translate into a significant gain in the computational cost, then the proposed strategy has a strong potential for reducing the global computational cost of the complete solution of the linear system.

4.2. Preconditioning. In the experiments proposed here, we have focused on the influence of perturbations of the matrix A (i.e., on inexact matrix-vector products) on the convergence of GMRES, regardless of the preconditioner. As a matter of fact, our strategy makes use of the residual of the original system $Ax = b$. However, in the case of left preconditioning, for instance, this residual may not be readily available: only the preconditioned residual would appear naturally in the algorithm. In such a case, it may be more appropriate to base the relaxation strategy on the preconditioned matrix, rather than on the original matrix.

4.3. Scaling issues. As mentioned in section 2.2, the relaxation strategy is based on the choice $\varepsilon_k = \min(\alpha_{k-1}\eta, 1)$, where $\alpha_{k-1} = 1/\min(\|r_{k-1}\|, 1)$ retains only an absolute information (the residual) from the outer process. Clearly this strategy suffers from the drawback of being scaling-dependent. Indeed, scaling the linear system $Ax = b$ by a constant will not change the convergence of GMRES but would definitely affect the relaxation strategy. It is therefore desirable to design a scaling-independent strategy. For instance, what would happen if one uses the backward error $\|r_{k-1}\|/(\|A\|\,\|x_{k-1}\|)$ instead of the residual $\|r_{k-1}\|$ alone? The corresponding strategy would be defined by

$$\varepsilon'_k = \min(\alpha'_{k-1}\eta, 1) \text{ with } \alpha'_{k-1} = \frac{1}{\min\left(\frac{\|r_{k-1}\|}{\|A\|\|x_{k-1}\|}, 1\right)}.$$

Surprisingly, this idea, which would seem natural a priori, does not lead to good results. Indeed, with such a choice, the convergence of the outer process is significantly delayed or even impeached (see Figures 4.1 and 4.2). Again, changing ε_k into ε'_k may be seen as another possible variant of the relaxation strategy if one also plays with the

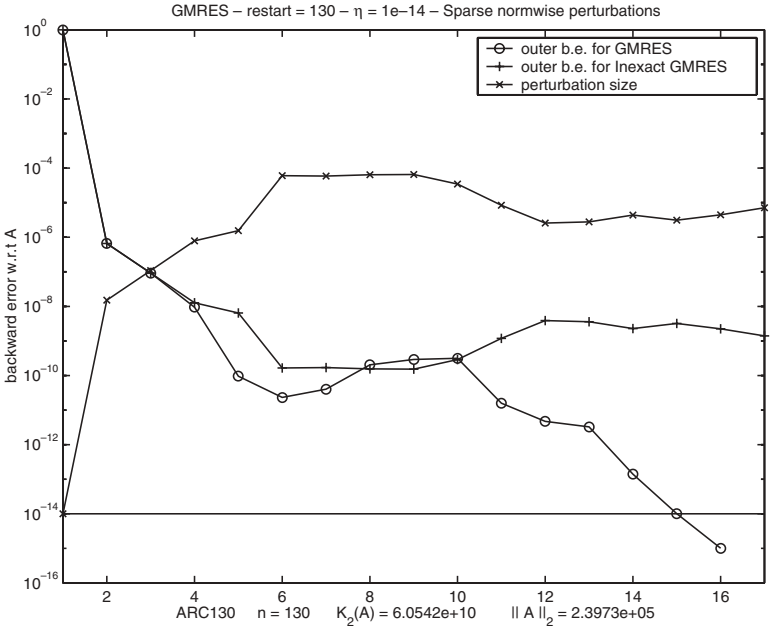


FIG. 4.1. *GMRES with inexact matrix-vector products.* `ARC130`. $\eta = 10^{-14}$. *Relaxation strategy with* ε'_k.

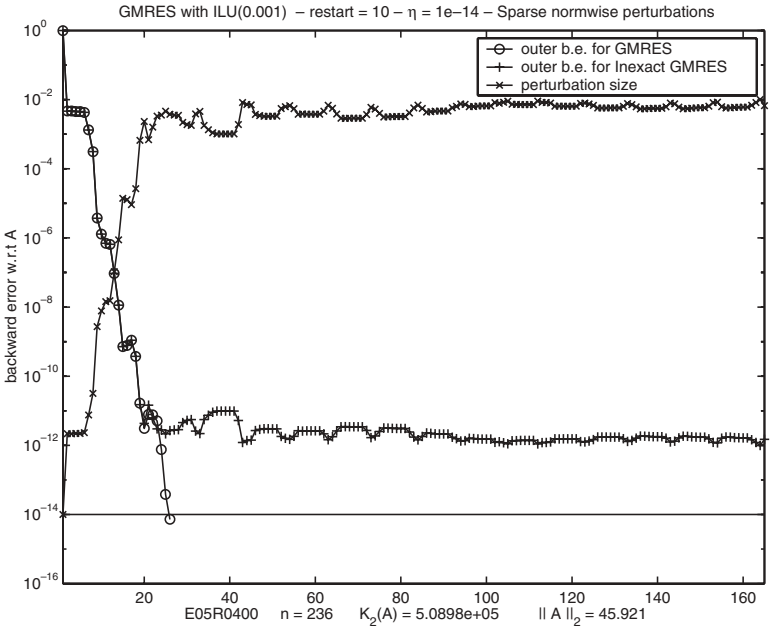


FIG. 4.2. *GMRES with inexact matrix-vector products.* `E05R0400`. $\eta = 10^{-14}$. *Relaxation strategy with* ε'_k.

choice of the parameter η, the targeted tolerance, and the stopping criterion, which can be any multiple of η. However, this means that further work remains to be done in order to obtain a deeper interpretation of the relaxation strategy, and why it seems to work so well.

4.4. Dependency on the starting vector. The dependency of the proposed relaxation strategy on the choice of the starting vector has also been explored. In practice we found that changing the initial vector, and thus the initial residual, would have no more effect on the global convergence than one would observe without using a relaxation scheme.

5. Conclusion and perspectives. Golub, Zhang, and Zha have been among the first to expose the robustness of Krylov schemes to variable inaccuracies in their work on inner-outer Lanczos processes in [13]. The inner process solves a linear least-squares problem with a tolerance τ_j which varies at each outer step j of the Lanczos algorithm. The authors discuss a way to deduce the sequence $\{\tau_j\}$ from the eigenvector associated with the smallest eigenvalue of the Lanczos tridiagonal matrix. It is noted that the sequence $\{\tau_j\}$ can grow (implying a lesser inner accuracy) and yet the outer process converges and the overall computation is cheaper.

Similarly, our experiments clearly demonstrate that Krylov methods are robust to inexact matrix-vector products, provided an appropriate strategy (of type (2.1), for example) is applied. In particular, the first vectors of the Krylov space need to be computed with full accuracy, while this constraint can be relaxed further on. It is remarkable that the Krylov process still converges while the Krylov vectors are significantly perturbed. In the case of linear systems, we have proposed a practical way (chosen amongst possible others for its large scope of good performance) to control the inner accuracy: the relaxation strategy indexes the accuracy of the kth Krylov vector (in terms of its backward error) on the reciprocal of the residual of the current iterate. Since inexact matrix-vector products induce an overhead in terms of outer iterations, it is crucial to see whether this overhead is compensated at the global level by the reduction of the cost of the inner level. Only then will a complete comparison of different relaxation schemes be possible.

This paper sets up a framework, inspired by the backward error analysis, which seems extremely promising for future investigations on the robustness of Krylov methods. Although essentially experimental, this work and the applications performed on eigenproblems [3] and on domain decomposition techniques with Giraud [4] seem to have captured the interest of engineers and researchers since the first time it was presented [1]. Among the practical uses of this relaxation scheme, we wish to cite the work of [21] and [15]. More recently, a few papers have shown significant progress in building the basis for a more theoretical explanation of the phenomena explored here: [10, 18, 20] show sufficient conditions for convergence of Krylov schemes under a relaxation scheme indexed on the reciprocal of the residual. Not surprisingly, these sufficient conditions on the perturbation size involve factors such as the condition number of the computed Hessenberg matrix or the matrix itself. Further work should be performed in order to check whether these conditions provide the full explanation for the robustness of Krylov methods observed in practice. Once their estimation using quantities readily available in the inner-outer schemes has been worked out carefully, these promising results may lead to more refined and more efficient relaxation strategies.

Appendix. Some convergence plots for inexact Krylov methods.

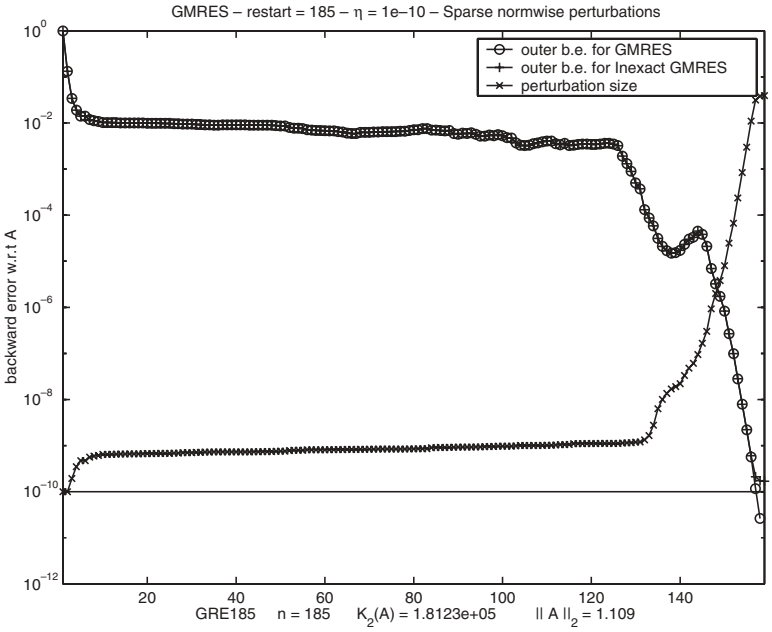


FIG. A.1. *GMRES with inexact matrix-vector products.* `GRE185`. $\eta = 10^{-10}$.

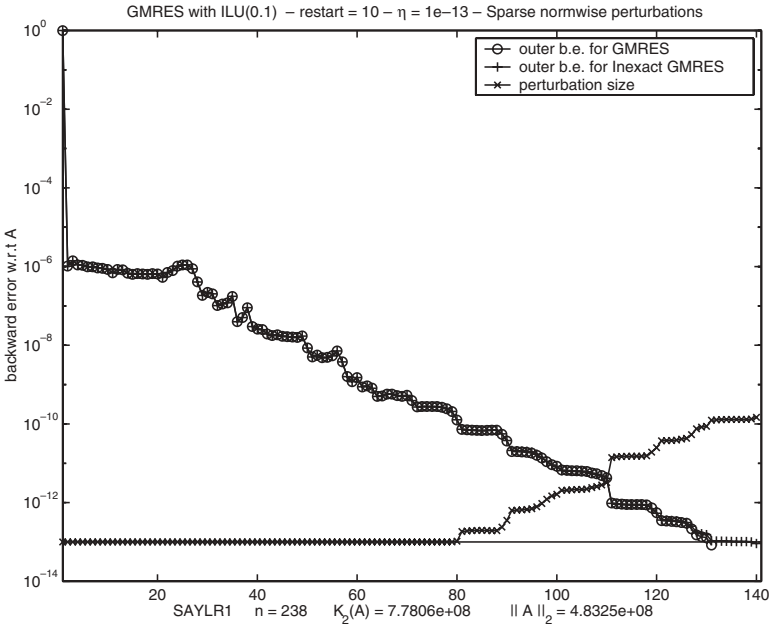


FIG. A.2. *GMRES*(m) *with inexact matrix-vector products.* `SAYLR1`. $\eta = 10^{-13}$, $m = 10$.

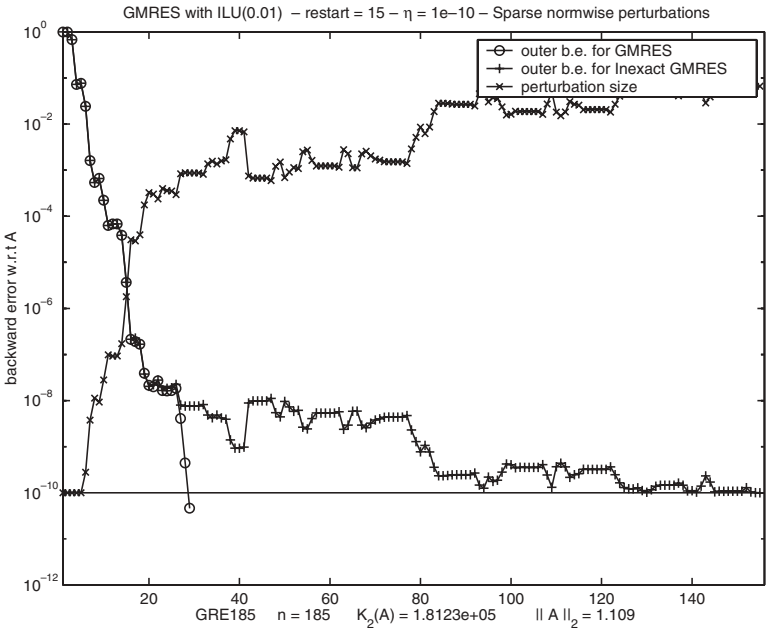


FIG. A.3. *GMRES*(m) *with inexact matrix-vector products.* `GRE185`. $\eta = 10^{-10},\ m = 15$.

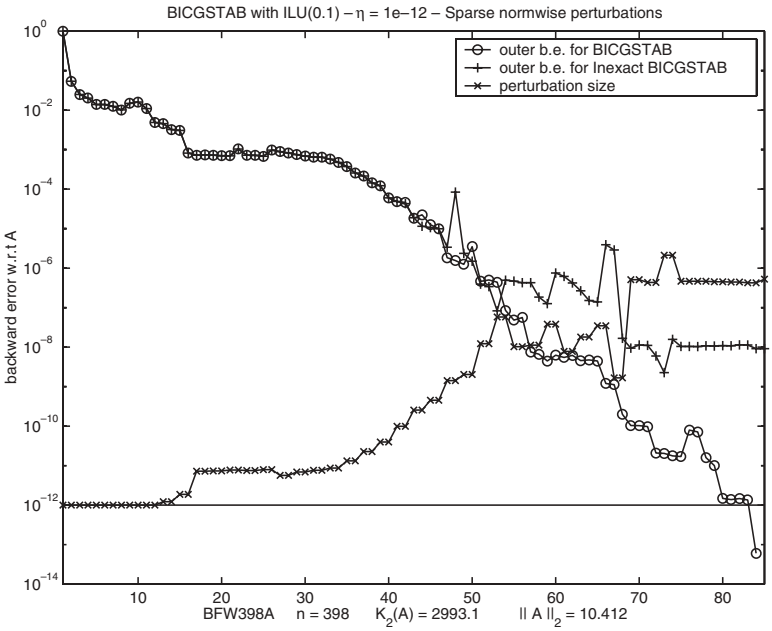


FIG. A.4. *BiCGStab with inexact matrix-vector products.* `BFW398A`. $\eta = 10^{-12}$.

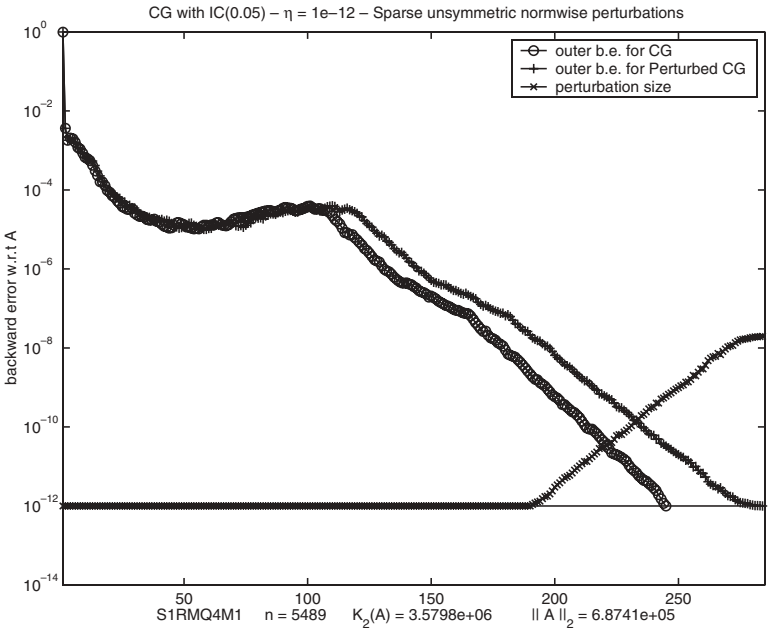


FIG. A.5. *CG with inexact matrix-vector products.* `S1RMQ4M1`. $\eta = 10^{-12}$.

Acknowledgments. The authors are indebted to Prof. F. Chaitin-Chatelin for insightful comments. The authors wish to acknowledge J.-C. Bergès from CNES (Centre National d'Etudes Spatiales) for the support granted for this work.

The authors are particularly grateful to L. Giraud (CERFACS) and S. Gratton (CERFACS) for their precious help, encouragement, support, and technical work, without which the final revision of this paper could not have been accomplished.

Finally, the authors would like to thank Prof. Gene Golub and the referees for their support and the interest they have shown in this experimental work. The very valuable remarks, questions, and comments given by the referees and in particular Anne Greenbaum have considerably affected the presentation and the perspective of the present work.

REFERENCES

[1] *Industrial Days at CERFACS on Inner-Outer Iterations*, 2000, http://www.cerfacs.fr/algor/iter2000.html.

[2] A. BOURAS AND V. FRAYSSÉ, *A Relaxation Strategy for Inexact Matrix-Vector Products for Krylov Methods*, Technical report TR/PA/00/15, CERFACS, Toulouse, France, 2000.

[3] A. BOURAS AND V. FRAYSSÉ, *A Relaxation Strategy for the Arnoldi Method in Eigenproblems*, Technical report TR/PA/00/16, CERFACS, Toulouse, France, 2000.

[4] 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, Toulouse, France, 2000.

[5] R. S. DEMBO, S. C. EISENSTAT, AND T. STEIHAUG, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[6] I. S. DUFF, R. G. GRIMES, AND J. G. LEWIS, *User's Guide for the Harwell-Boeing Sparse Matrix Collection*, Technical report TR-PA-92-86, CERFACS, Toulouse, France, 1992.

[7] S. C. EISENSTAT AND H. F. WALKER, *Globally convergent inexact Newton methods*, SIAM J. Optim., 4 (1994), pp. 393–422.

[8] E. Giladi, G. H. Golub, and J. B. Keller, *Inner and outer iterations for the Chebyshev algorithm*, SIAM J. Numer. Anal., 35 (1998), pp. 300–319.
[9] L. Giraud, *Private communication*, 1999.
[10] L. Giraud, S. Gratton, and J. Langou, *A Note on Relaxed and Flexible GMRES*, Technical report TR/PA/04/41, CERFACS, Toulouse, France, 2004.
[11] G. H. Golub and M. L. Overton, *The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems*, Numer. Math., 53 (1988), pp. 571–593.
[12] G. H. Golub and Q. Ye, *Inexact preconditioned conjugate gradient method with inner-outer iteration*, SIAM J. Sci. Comput., 21 (1999), pp. 1305–1320.
[13] G. H. Golub, Z. Zhang, and H. Zha, *Large sparse symmetric eigenvalue problems with homogeneous linear constraints: The Lanczos process with inner-outer iterations*, Linear Algebra Appl., 309 (2000), pp. 289–306.
[14] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Frontiers Appl. Math. 17, SIAM, Philadelphia, 1997.
[15] K. Mer-Nkonga and F. Collino, *The fast multipole method applied to a mixed integral system for time-harmonic Maxwell's equations*, in Proceedings of the European Symposium on Numerical Methods in Electromagnetics, B. Michielsen and F. Decavèle, eds., 2002, pp. 121–126.
[16] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS, Boston, 1996.
[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] 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.
[19] P. Smit and M.-H.-C. Paardekooper, *The effects of inexact solvers in algorithms for symmetric eigenvalue problems*, Linear Algebra and Appl., 287 (1999), pp. 337–357.
[20] 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.
[21] J. S. Warsa, M. Benzi, T. Wareing, and J. Morel, *Preconditioning of a mixed discontinuous finite element method for radiation diffusion*, Numer. Linear Algebra Appl., 11 (2004), pp. 795–811.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 3, pp. 679–685

SOLUTION FORM AND SIMPLE ITERATION OF A NONSYMMETRIC ALGEBRAIC RICCATI EQUATION ARISING IN TRANSPORT THEORY*

LIN-ZHANG LU†

Abstract. We are interested in computing the minimal positive solution of a nonsymmetric algebraic Riccati equation arising in transport theory. We show that this computation can be done via computing only the minimal positive solution of a vector equation, which is derived from the special form of solutions of the Riccati equation. A simple iterative method is presented for solving the vector equation. The simple iteration is much more efficient than the Gauss–Jacobi method presented by Juang in [*Linear Algebra Appl.*, 230 (1995), pp. 89–100] for the Riccati equation. The symmetric case and bounds of the minimal positive solution are also considered. Numerical experiments are given.

Key words. nonsymmetric algebraic Riccati equations, simple iteration, minimal positive solution

AMS subject classifications. 15A24, 65F10, 82C70

DOI. 10.1137/S0895479801397275

1. Introduction.

In this paper we are interested in iteratively solving the following algebraic Riccati equation arising in transport theory (see [6], [7] and the references cited therein):

$$XCX - XE - AX + B = 0, \tag{1}$$

where $A, B, C, E \in R^{n\times n}$ are given by

$$A = \Delta - eq^T,\ B = ee^T,\ C = qq^T,\ E = D - qe^T. \tag{2}$$

Here $e = (1, 1, \ldots, 1)^T,\ q = (q_1, q_2, \ldots, q_n)^T$ with $q_i = \frac{c_i}{2\omega_i}$,

$$\begin{cases} \Delta = \mathrm{diag}(\delta_1, \delta_2, \ldots, \delta_n) & \text{with } \delta_i = \frac{1}{c\omega_i(1+\alpha)}, \\ D = \mathrm{diag}(d_1, d_2, \ldots, d_n) & \text{with } d_i = \frac{1}{c\omega_i(1-\alpha)}, \end{cases} \tag{3}$$

and $0 < c \le 1,\ 0 \le \alpha < 1,\quad 0 < \omega_n < \cdots < \omega_2 < \omega_1 < 1,$

$$\sum_{i=1}^{n} c_i = 1,\ c_i > 0, \quad i = 1, 2, \ldots, n.$$

It has been shown in [6] and [7] that (1) has positive solutions (in the componentwise sense). Since only the minimal positive solution is physically meaningful, some

*Received by the editors October 30, 2001; accepted for publication (in revised form) by A. C. M. Ran June 23, 2004; published electronically March 3, 2005. This work was supported by the National Natural Science Foundation of China through grant 10271099.

http://www.siam.org/journals/simax/26-3/39727.html

†Department of Information and Computational Mathematics, Xiamen University, Xiamen 361005, People's Republic of China (lzlu@xmu.edu.cn).

iterative methods have been developed for computing the minimal positive solution of (1) (see [6] and [7]). More general nonsymmetric algebraic Riccati equations have also been studied in [5] and [4].

However, the matrix equation (1) is in essence equivalent to a "vector equation." This fact is shown here and utilized to develop a simple but efficient iterative procedure to compute the minimal positive solution of (1) and estimate its bounds.

This paper is organized as follows. In section 2, we derive a vector equation from the special form of solutions of (1), and we show that the minimal positive solution of (1) can be computed via computing only the minimal positive solution of the vector equation. A simple but efficient iteration for the vector equation is given in this section. In section 3, we give some estimations for bounds of the minimal positive solution. In section 4, the symmetric case of (1) is discussed. Sections 5 and 6 give some numerical examples and conclusions, respectively.

In this paper, we will use the standard notation for matrices with nonnegative elements, as, e.g., used in [2]. Let $A = (a_{i,j}), B = (b_{i,j})$ be $n \times n$ matrices with real entries. The Hadamard product of A and B is defined by $A \circ B = (a_{i,j}b_{i,j})$.

2. Solution form and simple iteration. Rewrite (1) as

$$\Delta X + XD = (Xq + e)(q^T X + e^T), \tag{4}$$

and let

$$u = Xq + e, \quad v^T = q^T X + e^T. \tag{5}$$

Then any solution of (1) must be of the form

$$X = T \circ (uv^T) = (uv^T) \circ T, \tag{6}$$

where

$$T = (t_{i,j}) = \left(\frac{1}{\delta_i + d_j}\right), \tag{7}$$

$X = (x_{i,j})$, $u = (u_1, u_2, \ldots, u_n)^T$, and $v^T = (v_1, v_2, \ldots, v_n)$.

Remark 1. It has already been noted in [6] and [7] that the positive solutions of (1) are of the form (6).

To find the minimal positive solution of (1), we need to find proper positive vectors u and v in (6). For this, substituting (6) into (5), we obtain a "vector equation":

$$\begin{cases} u = u \circ (Pv) + e, \\ v = v \circ (\tilde{P}u) + e, \end{cases} \tag{8}$$

where

$$P = (p_{i,j}) = \left(\frac{q_j}{\delta_i + d_j}\right), \quad \tilde{P} = (\tilde{p}_{i,j}) = \left(\frac{q_j}{\delta_j + d_i}\right). \tag{9}$$

To get the positive vectors u and v from (8), we define a simple iteration for (8):

$$\begin{cases} u^{(k+1)} = u^{(k)} \circ (Pv^{(k)}) + e, \\ v^{(k+1)} = v^{(k)} \circ (\tilde{P}u^{(k)}) + e, \quad k = 0, 1, \ldots, \\ u^{(0)} = v^{(0)} = 0. \end{cases} \tag{10}$$

Then we have the following.

LEMMA 1. *For all* $0 < c \le 1$ *and* $0 \le \alpha < 1$, *the sequence* $\{(u^{(k)}, v^{(k)})\}$ *defined by* (10) *is strictly monotonically increasing, bounded above and thus converging.*

Proof. It is easy to show by the induction that $\{(u^{(k)}, v^{(k)})\}$ is strictly monotonically increasing. To show that $\{(u^{(k)}, v^{(k)})\}$ is bounded above, we observe the following Gauss–Jacobi (GJ) method defined by Juang (see [6] in the componentwise form):

$$\begin{cases} X^{(k+1)} = T \circ [(X^{(k)}q + e)(q^T X^{(k)} + e^T)], \quad k = 0, 1, \ldots, \\ X^{(0)} = 0. \end{cases} \tag{11}$$

An easy induction will give the following relation between iterations (10) and (11):

$$X^{(k)} = T \circ [u^{(k)}(v^{(k)})^T], \quad k = 0, 1, 2, \ldots. \tag{12}$$

Since $\{X^{(k)}\}$ is bounded above (see [6]), so is $\{(u^{(k)}, v^{(k)})\}$. Thus both the convergence of the simple iteration (10) and the existence of the positive solutions of (8) are assured. ∎

Now we can present our main result.

THEOREM 2. *The minimal positive solution* X^* *of* (1) *can be computed by*

$$X^* = T \circ (u^*(v^*)^T),$$

where (u^*, v^*) *is the limit of* $\{(u^{(k)}, v^{(k)})\}$ *defined by the simple iteration* (10).

Proof. Since the limit of $X^{(k)}$, produced by (11), is the minimal positive solution of (1) (see [6]), Theorem 1 thus follows by taking $k \to \infty$ on the two sides of (12). ∎

We know from Theorem 2 that the minimal positive solution of (1) can be obtained by computing the positive solution of (8). It should be easily understood that the vector equation (8) is simpler and should be easier to solve than the matrix equation (1). This is true to a certain extent. In fact, Lemma 1 and (12) make clear that iterations (10) and (11) can play the same role in finding the minimal positive solution of (1). However, the difference of the two iterations lies in the computational cost. The simple iteration needs about $4n^2$ flops (see [3] for the definition of the flops) for each iteration. By comparison, the GJ iteration needs about $6n^2$ flops for each iteration, n^2 flops more for the outer product of the two vectors and another n^2 flops more for the Hadamard product of the two matrices. So there are significant savings here. If we formed the matrix $X^{(k)}$ for each $k \ge 0$, then the sequence $\{X^{(k)}\}$ would be precisely the one obtained by the GJ method and there would be no saving over the GJ method. Thus it is clear that the simple iteration (10) is basically a more efficient implementation of the GJ method (11). Moreover, we have the error relationship:

$$X^* - X^{(k)} \le T \circ (u^*(v^* - v^{(k)})^T + (u^* - u^{(k)})(v^*)^T). \tag{13}$$

Remark 2. According to the discussions in [5] and [4], the convergence of the GJ method (and thus the simple iteration) is sublinear when $(\alpha, c) = (0, 1)$ and is linear when $(\alpha, c) \ne (0, 1)$.

3. Solution bounds. In this section we derive some bounds of the minimal positive solution of (1) by using the simple iteration (10).

For notational convenience, we write $w = (u^T, v^T)^T \in R^{2n}$ with $w_i = u_i$, $w_{n+i} = v_i$ $(i = 1, 2, \ldots, n)$ and define $g(w) = (g_1(w), g_2(w), \ldots, g_{2n}(w))^T$ with

$$g_i(w) = \begin{cases} \sum_{l=1}^{n} p_{i,l} w_{n+l} & \text{when } 1 \le i \le n, \\ \sum_{l=1}^{n} \tilde{p}_{i-n,l} w_l & \text{when } n < i \le 2n. \end{cases} \tag{14}$$

Then (8) and (10) can be rewritten as

$$w = w \circ g(w) + e, \tag{15}$$

$$w^{(k+1)} = w^{(k)} \circ g(w^{(k)}) + e, \quad w^{(0)} = 0, \quad k = 0, 1, \ldots. \tag{16}$$

Let w^* be the positive solution of (15) computed from (16), let $\gamma_1 = \min\{g_i(e)\}$ and let $\gamma_2 = \max\{g_i(e)\}$; then it is obvious that

$$w^* > e, \quad 0 < \gamma_1 e \le g(e) < \min(w_i^*) g(e) \le g(w^*) < e. \tag{17}$$

We have the following.

LEMMA 3.

$$\gamma_1 < \frac{1}{2}, \quad \gamma_2 < 1. \tag{18}$$

Proof. From (14) and $\sum_{i=1}^n c_i = 1$, we have, when $1 \le i \le n$,

$$g_i(e) = \sum_{l=1}^n p_{i,l} = \sum_{l=1}^n \frac{q_l}{\delta_i + d_l} = \frac{c}{2} \sum_{l=1}^n \frac{c_l \omega_i (1-\alpha^2)}{\omega_i(1+\alpha) + \omega_l(1-\alpha)} < \frac{c(1-\alpha)}{2};$$

when $n < i \le 2n$,

$$g_i(e) = \sum_{l=1}^n \tilde{p}_{i-n,l} = \sum_{l=1}^n \frac{q_l}{\delta_l + d_{i-n}} = \frac{c}{2} \sum_{l=1}^n \frac{c_l \omega_{i-n} (1-\alpha^2)}{\omega_{i-n}(1-\alpha) + \omega_l(1+\alpha)} < \frac{c(1+\alpha)}{2}.$$

Since $0 < c \le 1$ and $0 \le \alpha < 1$, (18) follows. □

Now we give lower and upper bounds for w^*.

LEMMA 4. *Let γ_1 and w^* be defined above. Then*

$$w^* \ge \frac{1-\gamma_1}{1-2\gamma_1} e. \tag{19}$$

Proof. Let $\{w^{(k)}\}$ be the sequence produced by the simple iteration (16). Then it is easy to verify that $w^{(1)} = e$ and $w^{(2)} = e + g(e) \ge e + \gamma_1 e$. Therefore,

$$w^* > w^{(2)} \ge (1 + \gamma_1) e. \tag{20}$$

Now we prove

$$w^* > \left[1 + \gamma_1 \sum_{j=0}^k (2\gamma_1)^j \right] e, \quad k = 0, 1, 2, \ldots, \tag{21}$$

by induction on k. Formula (20) shows that (21) is true for $k = 0$. Assume that (21) is true for k; then since $g(w^*) \ge \min(w_i^*) g(e)$, we have

$$w^* = w^* \circ g(w^*) + e \ge \left[\gamma_1 \left(1 + \gamma_1 \sum_{i=0}^k (2\gamma_1)^i \right)^2 + 1 \right] e > \left[1 + \gamma_1 \sum_{j=0}^{k+1} (2\gamma_1)^j \right] e.$$

We see that (21) is true for $k+1$. Thus

$$w^* \geq \left(1+\gamma_1\frac{1}{1-2\gamma_1}\right)e = \frac{1-\gamma_1}{1-2\gamma_1}e. \qquad \square$$

COROLLARY 5. *Let γ_1, γ_2, and w^* be defined as above; then we have*
(a) $\gamma_2 < 1-\frac{\gamma_1}{1-\gamma_1}$;
(b) $w_i^* < 1/\gamma_1$ *for some* i.
Proof. By Lemma 4, we have $e \leq \frac{1-2\gamma_1}{1-\gamma_1}w^*$ and thus

$$g_i(e) \leq \frac{1-2\gamma_1}{1-\gamma_1}g_i(w^*).$$

It follows from (17) that $\gamma_2 = \max\{g_i(e)\} < \frac{1-2\gamma_1}{1-\gamma_1}$. (a) is true.

On the other hand, since $\min\{w_i^*\} * g_i(e) \leq g_i(w^*) < 1$ and $g_i(e) \geq \gamma_1$, it follows that $\min\{w_i^*\} < 1/g_i(e) \leq 1/\gamma_1$. So (b) is true. $\square$

It follows from Corollary 5 that $1-2\gamma_1 > (1-\gamma_1)\gamma_1$. Thus $\gamma_1 < \frac{3-\sqrt{5}}{2}$. This bound for γ_1 is tighter than the one in Lemma 3.

As to upper bound of w^*, we can give only a rough estimation.

LEMMA 6. *If $\gamma_2 \leq 1/4$, then*

$$w^* \leq 2e \ \ \textit{or} \ \ \max(w_i^*) \leq 2. \tag{22}$$

Proof. Let $\{w^{(k)}\}$ be the sequence produced by the simple iteration (16). We first prove by the induction that $w^{(k)} \leq 2e$. Note that $w^{(1)} = e < 2e$. If $w^{(k)} \leq 2e$, then it follows by the definition of g_i (see (14)) that $g_i(w^{(k)}) \leq (\max(w_i^{(k)})) * g_i(e) \leq 2\gamma_2 \leq 1/2$. Thus (16) and the condition of the lemma give

$$w^{(k+1)} = w^{(k)} \circ g(w^{(k)}) + e \leq (\max(w_i^{(k)}) * \max(g_i(w^{(k)})) + 1)e \leq 2e.$$

Then $w^* \leq 2e$ since w^* is the limit of $\{w^{(k)}\}$. $\square$

Applying the results to bounds of the minimal positive solution of (1), we have the following.

THEOREM 7. *The minimal positive solution X of* (1) *has lower bound*

$$X \geq \left(\frac{1-\gamma_1}{1-2\gamma_1}\right)^2 T, \tag{23}$$

where T is as defined in (7). *If $\gamma_2 \leq 1/4$, then X has upper bound $X \leq 4T$.*

4. The symmetric Riccati equation. If $\alpha = 0$, then $\Delta = D = \operatorname{diag}(\frac{1}{c\omega_1}, \frac{1}{c\omega_2}, \ldots, \frac{1}{c\omega_n})$. Equation (1) becomes a symmetric Riccati equation:

$$DX + XD = (Xq+e)(q^TX + e^T). \tag{24}$$

Its solutions are symmetric and of the form

$$X = T \circ (uu^T), \ \text{ or } \ x_{i,j} = \frac{u_iu_j}{d_i+d_j}, \quad i,j = 1,2,\ldots,n. \tag{25}$$

Now $T = (t_{i,j}) = (\frac{1}{d_i+d_j})$ is a symmetric Cauchy matrix. The following proposition is immediate.

PROPOSITION 8. *For all* $0 < c \leq 1$, *any solution* X *of* (24) *is a positive semidefinite matrix.*

Proof. Lemma 3.7.1 of [1] showed that if both A and B are positive semidefinite matrices, then so is their Hadamard product $A \circ B$. Now we need to show only that both T and uu^T are positive semidefinite. Since $0 < d_1 < d_2 < \cdots < d_n$, the Cauchy matrix $T = (\frac{1}{d_i+d_j})$ is a positive definite matrix. It is clear that uu^T is a positive semidefinite matrix of rank one for any $u \neq 0$. □

Since $\alpha = 0$, we have $P = \tilde{P}$ in (9). The iteration (10) or (16) is simplified to

$$u^{(k+1)} = u^{(k)} \circ (Pu^{(k)}) + e \text{ with } u^{(0)} = 0, \quad k = 0, 1, \ldots. \tag{26}$$

For each iteration, (26) needs only half of the computational work of (10) or (16). Now we have the following result.

COROLLARY 9. *The minimal positive solution* $X = (x_{i,j})$ *of* (24) *has the lower bound*

$$x_{i,j} \geq \frac{(1-\gamma_1)^2}{(1-2\gamma_1)^2(d_i+d_j)}, \tag{27}$$

and there are some diagonal elements $x_{i,i}$ $(1 \leq i \leq n)$ *satisfying*

$$x_{i,i} < 1/(2\gamma_1^2 d_i). \tag{28}$$

If $\gamma_2 \leq 1/4$, *then* X *has the upper bound* $x_{i,j} \leq \frac{4}{d_i+d_j}$. *Now* $\gamma_1 = \min(Pe)_i$, $\gamma_2 = \max(Pe)_i$, *where* $P = (p_{i,j}) = (\frac{q_j}{d_i+d_j})$.

5. Numerical experiments. In the following examples, we let

$$r_k = ||R(X^{(k)})||_\infty = ||\Delta X^{(k)} + X^{(k)}D - (X^{(k)}q + e)(q^T X^{(k)} + e^T)||_\infty,$$

the residual of (4) or (1). GJ, SI will denote the iteration method decided by (11), (10), respectively.

Example 1. This example is used to test the time that the iteration methods GJ and SI need when the iterations converge.

We consider (1) for $n = 64$ and $n = 128$. As in Example 5.2 in [5], the constants c_i and ω_i are given by a numerical quadrature formula on the interval $[0, 1]$, which is obtained by dividing $[0, 1]$ into $n/4$ subinterval of equal length and applying a Gauss–Legendre quadrature with 4 nodes to each subinterval.

We have tested three values of (α, c)—$(0.5, 0.5)$, $(0.85, 0.1)$, and $(10^{-8}, 1-10^{-6})$—and recorded the iteration counts and times that iterations need to have $r_k/r_0 < \epsilon$ (ϵ is taken ranging from 10^{-2} to 10^{-12}) for GJ and SI. From the numerical experiment, we found that GJ and SI need nearly the same iteration counts, but the cpu time for SI is about 2/3 of that for GJ when n is large enough, e.g., no less than 32, and if the work needed to check the stopping criterion is negligible.

Example 2. Two examples are used to compare maximum and minimum elements of computed solution of (15) with the estimated bounds (19) and (22) given in section 3.

We first consider (1) for $n = 8$, and the constants c_i and ω_i are given by a numerical quadrature formula on the interval $[0, 1]$, which is obtained by applying a Gauss–Legendre quadrature with 8 nodes. The results are recorded in Table 1.

Second, we consider (1) for $n = 128$, with c_i and ω_i as given in Example 1. The results are given in Table 2. From Tables 1 and 2, we can see that the estimated

TABLE 1
Comparison: Computed solutions vs. estimated bounds, $n = 8$.

(α, c)	$(.01, .99)$	$(.05, .95)$	$(.1, .9)$	$(.2, .9)$	$(.6, .4)$	$(.7, .3)$
$\min(w_i^*)$	1.0174	1.0162	1.0148	1.0141	1.0038	1.0022
est. low.	1.0158	1.0150	1.0139	1.0133	1.0037	1.0022

TABLE 2
Comparison: Computed solutions vs. estimated bounds, $n = 128$.

(α, c)	$(0, 1)$	$(0.1, 0.9)$	$(0.2, 0.8)$	$(0.3, 0.7)$	$(0.4, 0.6)$	$(0.5, 0.5)$
$\min(w_i^*)$	1.0491	1.0342	1.0258	1.0192	1.0139	1.0095
est. low.	1.0328	1.0272	1.0219	1.0171	1.0127	1.0090
$\max(w_i^*)$	2.8517	1.8689	1.6132	1.4485	1.3267	1.2316
est. upp.	no	no	no	no	2	2
γ_2	0.3445	0.3212	0.2909	0.2548	0.2143	0.1710

lower bounds are very close to the least elements of the computed solutions. This shows that our estimation for the lower bound of the minimal positive solution is appropriate. But it can also be seen from Table 2 that our estimation for the upper bound is too rough. In particular, we have not covered the case of $1/4 < \gamma_2 < 1 - \frac{\gamma_1}{1-\gamma_1}$ (see Corollary 5(a)).

6. Conclusions. By analyzing the special form of solutions of (1), we showed that any solution of the nonsymmetric algebraic Riccati equation is only related to two vectors and given an equation that the two vectors need to satisfy. The vector equation can be utilized to study and compute the minimal positive solution of (1). A simple iterative method was presented for the vector equation that is more efficient than the GJ method presented by Juang in [6] for the Riccati equation. The symmetric case of the Riccati equation was also considered. By using the simple iteration, we also gave some bounds of the minimal positive solution. Numerical experiments were given to verify the results. However, there are still some problems for future work. For example, since the convergence of the simple iteration is only sublinear, it is of interest to study the use of some methods of higher order convergence, e.g., Newton's method, to solve (15).

Acknowledgments. The author wishes to thank the referees for their patient readings, detailed comments, and some corrections which helped to improve the presentation of the paper.

REFERENCES

[1] R. B. BAPAT AND T. E. S. RAGHAVAN, *Nonnegative Matrices and Applications*, Cambridge University Press, Cambridge, UK, 1997.
[2] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, Philadelphia, 1994.
[3] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[4] C. H. GUO, *Nonsymmetric algebraic Riccati equations and Wiener–Hopf factorization for M-matrices*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 225–242.
[5] 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.
[6] J. JUANG, *Existence of algebraic matrix Riccati equations arising in transport theory*, Linear Algebra Appl., 230 (1995), pp. 89–100.
[7] J. JUANG AND W.-W. LIN, *Nonsymmetric algebraic Riccati equations and Hamiltonian-like matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 228–243.
[8] R. S. VARGA, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 3, pp. 686–705

THE THEORY 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 matrix factorization. By using paths instead of edges to define the tree, we generalize this structure to unsymmetric matrices while retaining many of its properties. If we use a tree traversal to reorder a matrix into a bordered block triangular form, the structure has further desirable properties relevant to a sparse *LU* factorization of the reordered matrix. When pivoting is required for stability, the tree changes only locally if the choice of pivot is suitably restricted.

Key words. sparse matrix factorization, sparse LU decomposition

AMS subject classifications. 65F05, 65F50

DOI. 10.1137/S089547980240563X

1. Introduction.

Schreiber [20] defined the *elimination tree* of a sparse symmetric matrix. This structure has many important roles in sparse matrix factorization, including storage schemes, matrix reordering, symbolic and numerical factorization, and parallel elimination [18].

Gilbert and Liu [11] extended the elimination tree to study the sparse LU factorization of an unsymmetric matrix. Their generalization consists of a pair of *elimination dags* (directed acyclic graphs), which are the transitive reductions of the graphs of the lower and upper triangular factors.

In this paper we present a new generalization that uses paths instead of edges to define a single tree structure with many of the same properties as the elimination tree of a symmetric matrix. The approach is closely related to the use of path-symmetric reductions to speed up symbolic LU factorization [7].

The outline of the paper is as follows. In section 2 we introduce some graph notation and present the relevant background for sparse LU factorization.

In section 3 we generalize the notion of elimination tree from symmetric matrices to unsymmetric matrices by using paths instead of edges and characterize the ancestor-descendant relation for vertices in the resulting tree structure. We also relate subtrees of the elimination tree to strongly connected subgraphs of the original directed graph.

In section 4 we use the elimination tree to characterize the row and column structures of the factor matrices. In particular, we show that the structure of each row of L and each column of U is a pruned forest of the elimination tree.

In section 5 we explore the use of topological orderings and postorderings of the elimination tree to reorder a sparse unsymmetric matrix. The resulting *BBT postordering* gives a reordered matrix with a lower or upper bordered block triangular form.

*Received by the editors April 15, 2002; accepted for publication (in revised form) by E. Ng May 18, 2004; published electronically March 3, 2005.
http://www.siam.org/journals/simax/26-3/40563.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, Ontario, Canada M3J 1P3 (joseph@cs.yorku.ca). The research of this author was supported in part by the Natural Sciences and Engineering Research Council of Canada under grant A5509.

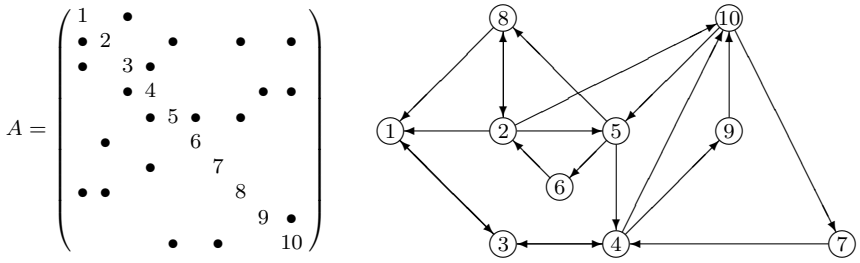


FIG. 1. *A sparse unsymmetric matrix and its directed graph.*

In section 6 we establish some important properties of matrices that have been reordered by upper BBT postorderings, showing that each row structure of L is a pruned subtree (instead of a pruned forest); that the elimination dag of L is the same as the elimination tree; that the tree is a depth-first tree of a certain depth-first traversal of the filled graph; and that it captures the data dependencies among the rows of the filled matrix. Similar results hold for lower BBT postorderings and U.

In section 7 we consider how pivoting for stability can affect the elimination tree. While a single row exchange can change the entire tree, we show that both delayed elimination and the restricted form of off-diagonal pivoting used in the unsymmetric multifrontal method have only a local effect on the tree structure.

In section 8 we give some concluding remarks and discuss potential applications of the elimination tree structure to other aspects of sparse LU factorization.

2. Background.

2.1. Graph notation. Let M be a sparse unsymmetric $n \times n$ matrix. The *directed graph* $G(M)$ of M is defined as follows: the vertex set is $X(M) = \{1, 2, \ldots, n\}$, and there is an edge from vertex r to vertex c (for $r \neq c$) if and only if the entry $m_{rc} \neq 0$. We shall use the notation $r \overset{M}{\longmapsto} c$ to indicate a directed edge from r to c. If the matrix M is clear from context, we shall sometimes use the abbreviated form $r \mapsto c$.

Furthermore, we shall use the notation $r \overset{M}{\Longrightarrow} c$ to indicate a *directed path*[1] from vertex r to vertex c in $G(M)$, and the notation $r \overset{M}{\Longrightarrow}_{\min} c$ (respectively, $r \overset{M}{\Longrightarrow}_{\max} c$) to indicate a directed path from r to c whose intermediate vertices (if any) lie in the subset $\{1, \ldots, m-1\}$, where $m = \min\{r, c\}$ (respectively, $m = \max\{r, c\}$). If the matrix M is clear from context, we shall sometimes use the abbreviated forms $r \Rightarrow c$, $r \Rightarrow_{\min} c$, and $r \Rightarrow_{\max} c$.

A set of vertices $S \subseteq X(M)$ induces a *subgraph* of $G(M)$ consisting of the vertices in S and all edges $x \mapsto y$ in $G(M)$ with $x, y \in S$. 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 $\{1, \ldots, m\}$ of $G(M)$ for $1 \leq m \leq n$.

Figure 1 contains a 10×10 unsymmetric matrix that will be used throughout the paper to illustrate various notions and properties. Note that $9 \Rightarrow 1$, but not $9 \Rightarrow_{\max} 1$, and that $6 \Rightarrow_{\max} 1$, but not $6 \Rightarrow_{\min} 1$.

[1] Paths and cycles need not be *simple*; that is, they may visit a vertex more than once.

$$A^+ = \begin{pmatrix} 1 & \bullet & & & & & & & & \\ \bullet & 2 & \circ & & \bullet & & & \bullet & & \bullet \\ \bullet & & 3 & \bullet & & & & & & \\ & & \bullet & 4 & & & & & \bullet & \bullet \\ & & & \bullet & 5 & \bullet & & \bullet & \circ & \circ \\ & \bullet & \circ & \circ & \circ & 6 & & \circ & \circ & \circ \\ & & & \bullet & & & 7 & & \circ & \circ \\ \bullet & \bullet & \circ & \circ & \circ & \circ & & 8 & \circ & \circ \\ & & & & & & & & 9 & \bullet \\ & & & & \bullet & \circ & \bullet & \circ & \circ & 10 \end{pmatrix}$$

FIG. 2. *The filled matrix for the matrix in Figure* 1.

2.2. The filled graph and the elimination dags. 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. The *filled matrix* of A is $A^+ = L + U - I$. The *filled graph* of A is the directed graph $G(A^+)$ of its filled matrix.

It is well known that fill-in can occur during the LU factorization of a sparse matrix; that is, there can be entries that are zero in the original matrix A but nonzero in the filled matrix A^+. The following "path-theorem" of Rose and Tarjan [19] characterizes the locations of the nonzero entries in A^+.

THEOREM 2.1 (see [19, Theorem 1]). *There exists an edge* $r \overset{A^+}{\longmapsto} c$ *in* $G(A^+)$ *if and only if there exists a path* $r \overset{A}{\Longrightarrow}_{\min} c$ *in* $G(A)$.

Figure 2 gives the filled matrix A^+ for the matrix A in Figure 1 with the 20 fills depicted by $\circ$. In the directed graph $G(A)$ there are paths

$$10 \overset{A}{\longmapsto} 5 \overset{A}{\longmapsto} 6 \overset{A}{\longmapsto} 2 \overset{A}{\longmapsto} 8 \quad \text{and} \quad 8 \overset{A}{\longmapsto} 1 \overset{A}{\longmapsto} 3 \overset{A}{\longmapsto} 4 \overset{A}{\longmapsto} 10$$

through vertices less than 8. By Theorem 2.1 there are edges $10 \overset{A^+}{\longmapsto} 8$ and $8 \overset{A^+}{\longmapsto} 10$ or, equivalently, $10 \overset{L}{\longmapsto} 8$ and $8 \overset{U}{\longmapsto} 10$.

Theorem 2.1 characterizes the edges in the filled graph $G(A^+)$ in terms of *fill paths* $r \overset{A}{\Longrightarrow}_{\min} c$ in the original graph $G(A)$. Alternatively, we can characterize the nonzero structures of the lower and upper triangular factors in terms of the edges in $G(A)$ and the set of paths in $G(U)$ and $G(L)$, respectively: the structure of the rth row of L is given by

$$\{j \mid j < r \text{ and } r \overset{A}{\longmapsto} k \overset{U}{\Longrightarrow} j \text{ for some } k < r\},$$

and the structure of the cth column of U is given by

$$\{i \mid i < c \text{ and } i \overset{L}{\Longrightarrow} k \overset{A}{\longmapsto} c \text{ for some } k < c\}$$

(see [8, Theorem 3.2]).

The directed graphs $G(L)$ and $G(U)$ are *dags* (directed acyclic graphs) since they cannot have cycles. The *elimination dags* $G(L^o)$ and $G(U^o)$ of A are the transitive reductions[2] of $G(L)$ and $G(U)$, respectively, [11]. Since transitive reduction preserves the set of paths in a graph, the row structures of L can also be characterized in terms of the set of paths in $G(U^o)$ instead of the set of paths in $G(U)$. A similar observation applies to the column structures of U.

[2] A *transitive reduction* of a directed graph $G(M)$ is a subgraph $G(M^o)$ of $G(M)$ such that there is a path from vertex x to vertex y in $G(M)$ if and only if there is a path from x to y in $G(M^o)$, and no subgraph with this property has fewer edges. The transitive reduction of a dag is unique [1].

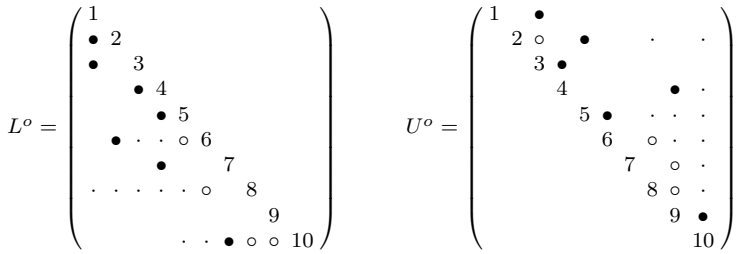


FIG. 3. *The matrix structures of the elimination dags for the matrix in Figure* 1.

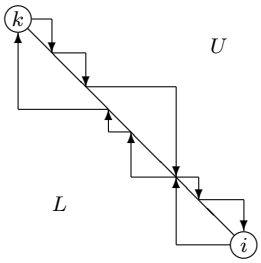


FIG. 4. *A matrix view of the cycle* $i \xrightarrow{L} k \xrightarrow{U} i$.

Figure 3 shows the matrix structures L^o and U^o of the elimination dags for the filled matrix in Figure 2. Entries removed from L and U are depicted by $\cdot$.

3. The elimination tree of an unsymmetric matrix. The pair of elimination dags can be viewed as a generalization of the elimination tree to unsymmetric matrices since they characterize the nonzero structures of the factor matrices in the same way that the elimination tree does in the symmetric case [11]. In this section we provide a new generalization that gives rise to a single tree/forest structure.

3.1. The definition. If the matrix A is symmetric, so is the filled matrix A^+. The elimination tree/forest of A is defined in terms of the function[3]

$$\text{FNZ}(k) = \min\{i \,|\, \ell_{ik} \neq 0\},$$

where vertex $p = \text{FNZ}(k)$ is the parent of vertex k if $\text{FNZ}(k) < \infty$. Since A^+ is also symmetric, we may choose $U = L^T$ (structurally if not numerically). Then the condition $\ell_{ik} \neq 0$ (or $i \xmapsto{L} k$) is equivalent to $i \xmapsto{L} k \xmapsto{U} i$ (or $i \xmapsto{L} k \xmapsto{L^T} i$).

When A is unsymmetric, we generalize this notion using paths instead of edges. The *elimination tree/forest* $T(A)$ of the unsymmetric matrix A is defined in terms of the function

$$\text{FPNZ}(k) = \min\{i \,|\, i \xrightarrow{L} k \xrightarrow{U} i\},$$

where vertex $p = \text{FPNZ}(k)$ is the parent of vertex k if $\text{FPNZ}(k) < \infty$. Figure 4 gives a matrix view of the cycle $i \xrightarrow{L} k \xrightarrow{U} i$ used in the definition of $\text{FPNZ}(k)$.

Figure 5 gives the elimination tree $T(A)$ for the matrix A in Figure 1. Since vertices 6, 8, and 10 are the only ones with cycles of the form $x \xrightarrow{L} 2 \xrightarrow{U} x$, namely,

$$6 \xmapsto{L} 2 \xmapsto{U} 5 \xmapsto{U} 6, \quad 8 \xmapsto{L} 2 \xmapsto{U} 8, \quad \text{and} \quad 10 \xmapsto{L} 6 \xmapsto{L} 2 \xmapsto{U} 10,$$

[3]If the set $\{i \,|\, \ell_{ik} \neq 0\}$ is empty, then the minimum is taken to be ∞.

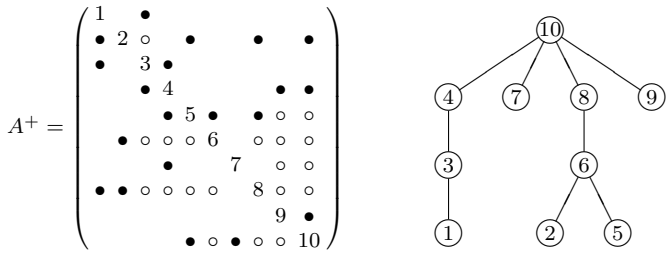


FIG. 5. *The filled matrix and elimination tree for the matrix in Figure* 1.

we have $\text{FPNZ}(2) = 6$.

It should be clear that this definition of elimination tree for unsymmetric matrices generalizes that for symmetric matrices. Indeed, if the matrix A is symmetric, then $\text{FNZ}(*) = \text{FPNZ}(*)$ and both functions give the same tree.

By definition we have $\text{FPNZ}(n) = \infty$. In general there may be other vertices with $\text{FPNZ}(k) = \infty$, in which case the elimination structure defined by $\text{FPNZ}(k)$ is a forest. (Corollary 3.7 characterizes when this can occur.) However, we shall still refer to this structure as an *elimination tree*.

3.2. A characterization of the ancestor-descendant relation. To explore the properties of the elimination tree $T(A)$, we characterize its ancestor-descendant relation in terms of certain cycles. We first extend the "path-theorem" Theorem 2.1 to characterize paths in the filled graph $G(A^+)$ in terms of paths in the original graph $G(A)$ or in the factor graphs $G(L)$ and $G(U)$.

THEOREM 3.1. *The following conditions are equivalent:*

(a) *There is a path* $r \overset{A^+}{\Longrightarrow}_{\max} c$ *in* $G(A^+)$.

(b) *There is a path* $r \overset{A}{\Longrightarrow}_{\max} c$ *in* $G(A)$.

(c) *Either there is a path* $r \overset{L}{\Longrightarrow} c$ *(if* $r \geq c$*) or there is a path* $r \overset{U}{\Longrightarrow} c$ *(if* $r \leq c$*).*

Proof. (a) implies (b). Assume that $r \overset{A^+}{\Longrightarrow}_{\max} c$. By Theorem 2.1, if $x \overset{A^+}{\longmapsto} y$ is an edge on that path, then there exists a path $x \overset{A}{\Longrightarrow}_{\min} y$ in $G(A)$. If we replace each edge by its corresponding path, we get a path $r \overset{A}{\Longrightarrow}_{\max} c$.

(b) implies (c). Assume that $r \overset{A}{\Longrightarrow}_{\max} c$ with $r \geq c$. Let x be the vertex on this path other than r with the largest label. Then $x < r$ and we can decompose the path into subpaths $r \overset{A}{\Longrightarrow}_{\min} x$ and $x \overset{A}{\Longrightarrow}_{\max} c$, where the latter will be empty if $x = c$. By Theorem 2.1 the edge $r \overset{L}{\longmapsto} x$ exists, and by a simple induction argument there exists a path $x \overset{L}{\Longrightarrow} c$. Pasting these together, we get a path $r \overset{L}{\Longrightarrow} c$. The case $r \leq c$ is similar.

(c) implies (a). Assume that $r \overset{L}{\Longrightarrow} c$. This path can be written as $r \overset{L}{\Longrightarrow}_{\max} c$, since L is lower triangular, and as $r \overset{A^+}{\Longrightarrow}_{\max} c$, since $G(L)$ is a subgraph of $G(A^+)$. The case $r \overset{U}{\Longrightarrow} c$ is similar. ∎

Using the equivalence above, we can give an alternative definition of $\text{FPNZ}(*)$:

$$\text{FPNZ}(k) = \min\{i \mid i > k \text{ and } i \overset{A}{\Longrightarrow}_{\max} k \overset{A}{\Longrightarrow}_{\max} i\}. \tag{1}$$

We now use these results to characterize the ancestor-descendant relation.

THEOREM 3.2. *The following conditions are equivalent:*

(a) *Vertex* q *is an ancestor of vertex* k *in the elimination tree* $T(A)$.

(b) *There is a cycle* $q \overset{L}{\Longrightarrow} k \overset{U}{\Longrightarrow} q$.

(c) $q > k$*, and there is a cycle* $q \overset{A}{\Longrightarrow}_{\max} k \overset{A}{\Longrightarrow}_{\max} q$ *in* $G(A)$.

Proof. (a) implies (b). Assume that q is an ancestor of k in $T(A)$. If vertex p is the parent of k, then there exists a cycle $p \xrightarrow{L} k \xrightarrow{U} p$. If $p = q$, we are done. Otherwise q must be an ancestor of p, and by a simple induction argument there exists a cycle $q \xrightarrow{L} p \xrightarrow{U} q$. Pasting pieces of these cycles together, we get the cycle

$$q \xrightarrow{L} p \xrightarrow{L} k \xrightarrow{U} p \xrightarrow{U} q.$$

(b) implies (c). Assume that there exists a cycle $q \xrightarrow{L} k \xrightarrow{U} q$. Since L is lower triangular, we have $q > k$. By Theorem 3.1 there exist paths $q \xrightarrow{A}_{\max} k$ and $k \xrightarrow{A}_{\max} q$. Pasting these together gives the result.

(c) implies (a). Let k be the vertex with the largest label for which $q > k$ and there exists a cycle $q \xrightarrow{A}_{\max} k \xrightarrow{A}_{\max} q$, but q is not an ancestor of k in $T(A)$. By (1) we must have $\text{FPNZ}(k) < q$. Thus k has a parent $p < q$ in $T(A)$, and there exists a cycle $p \xrightarrow{A}_{\max} k \xrightarrow{A}_{\max} p$. Pasting pieces of these cycles together, we get the cycle

$$q \xrightarrow{A}_{\max} k \xrightarrow{A}_{\max} p \xrightarrow{A}_{\max} k \xrightarrow{A}_{\max} q,$$

which can also be written as $q \xrightarrow{A}_{\max} p \xrightarrow{A}_{\max} q$. But q cannot be an ancestor of p either, which contradicts the definition of k. ∎

Theorem 3.2 characterizes the ancestor-descendant relation in $T(A)$ in terms of paths in $G(L)$, $G(U)$, and $G(A)$. For example, in the elimination tree in Figure 5, vertex 8 is an ancestor of vertex 5 and we have the cycles

$$8 \xmapsto{L} 5 \xmapsto{U} 8 \quad \text{and} \quad 8 \xmapsto{A} 2 \xmapsto{A} 5 \xmapsto{A} 8,$$

and vertex 10 is an ancestor of vertex 1 and we have the cycles

$$10 \xmapsto{L} 8 \xmapsto{L} 1 \xmapsto{U} 3 \xmapsto{U} 4 \xmapsto{U} 10 \quad \text{and} \quad 10 \xmapsto{A} 5 \xmapsto{A} 8 \xmapsto{A} 1 \xmapsto{A} 3 \xmapsto{A} 4 \xmapsto{A} 10.$$

3.3. Strongly connected properties of subtrees. A directed graph is said to be *strongly connected* if there exists a path between any pair of vertices; that is, there is a cycle connecting them. In this section we establish the main property of the elimination tree: that each subtree is a strongly connected component of a larger subgraph.[4] We begin with a characterization of the ancestor-descendant relation in terms of strongly connected components. Recall that $G_m(A)$ is the subgraph $\{1, \ldots, m\}$ of $G(A)$.

THEOREM 3.3. *Vertex q is an ancestor of vertex k in the elimination tree $T(A)$ if and only if $q > k$ and q and k belong to the same strongly connected component of the subgraph $G_q(A)$ of $G(A)$.*

Proof. Assume that q is an ancestor of k. Then by Theorem 3.2 we have $q > k$ and there exists a cycle $q \xrightarrow{A}_{\max} k \xrightarrow{A}_{\max} q$ in $G(A)$. Since this cycle lies in $G_q(A)$, the vertices q and k must belong to the same strongly connected component of that subgraph.

Conversely, assume that $q > k$ and q and k belong to the same strongly connected component of $G_q(A)$. Then there exists a cycle $q \Rightarrow k \Rightarrow q$ in that subgraph, which can be written as $q \xrightarrow{A}_{\max} k \xrightarrow{A}_{\max} q$. By Theorem 3.2 vertex q is an ancestor of vertex k. ∎

[4] In the symmetric case each subtree of the elimination tree corresponds to a *connected* subgraph of the original undirected graph.

This result can be used to characterize[5] the parent-child relation in the elimination tree.

COROLLARY 3.4. *Vertex p is the parent of vertex k in the elimination tree $T(A)$ if and only if p is the first vertex after k such that k and p belong to the same strongly connected component of the subgraph $G_p(A)$ of $G(A)$.*

For any vertex k, let $\mathcal{T}[k]$ denote both the subtree of $T(A)$ rooted at k and the set of vertices in this subtree, and let $\bar{\mathcal{A}}[k]$ denote the set of vertices in the graph $G(A)$ that are *not* proper ancestors of k in $T(A)$. We shall also let $\mathcal{T}[k]$ and $\bar{\mathcal{A}}[k]$ refer to the subgraphs of $G(A)$ that they induce. For example, the subtree $\mathcal{T}[8]$ of the elimination tree in Figure 5 contains the vertices 2, 5, 6, and 8 and corresponds to a subgraph of the graph in Figure 1, and $\bar{\mathcal{A}}[6] = \{1, 2, 3, 4, 5, 6, 7, 9\} = X(A) \setminus \{8, 10\}$ corresponds to another subgraph.

THEOREM 3.5. *The subgraph $\mathcal{T}[k]$ of $G(A)$ is a strongly connected component of the subgraph $\bar{\mathcal{A}}[k]$ of $G(A)$.*

Proof. By Theorem 3.3 every vertex $x \neq k$ in $\mathcal{T}[k]$ is in the same strongly connected component of $G_k(A)$ as k. But we have $\{1, \ldots, k\} \subseteq \bar{\mathcal{A}}[k]$, so k and x must belong to the same strongly connected component of $\bar{\mathcal{A}}[k]$.

It remains to prove that no other vertex belongs to this component. Assume otherwise and let y be the vertex with the largest label that is in the component but not in $\mathcal{T}[k]$. Then there exists a cycle $y \Rightarrow k \Rightarrow y$ in the subgraph $\bar{\mathcal{A}}[k]$. By the choice of y the intermediate vertices in this cycle must be either in $\mathcal{T}[k]$ (and thus less than k) or less than y. Thus it can be written as the cycle $y \Rightarrow_{\max} k \Rightarrow_{\max} y$ in the subgraph $\bar{\mathcal{A}}[k]$, and as $y \overset{A}{\Longrightarrow}_{\max} k \overset{A}{\Longrightarrow}_{\max} y$, since $\bar{\mathcal{A}}[k]$ is a subgraph of $G(A)$. By Theorem 3.2, if $y > k$, then y is an ancestor of k in the elimination tree, which contradicts the assumption that $y \notin \bar{\mathcal{A}}[k]$. Similarly, if $y < k$, then k is an ancestor of y in the elimination tree, which contradicts the assumption that $y \in \mathcal{T}[k]$. □

Since $\mathcal{T}[k]$ is a strongly connected component of $\bar{\mathcal{A}}[k]$, it is also a strongly connected component of any subgraph Y with $\mathcal{T}[k] \subseteq Y \subseteq \bar{\mathcal{A}}[k]$. Since the subgraph $G_k(A)$ containing $\{1, \ldots, k\}$ satisfies this condition, the following result is immediate.

COROLLARY 3.6. *The subgraph $\mathcal{T}[k]$ of $G(A)$ is a strongly connected component of the subgraph $G_k(A)$ of $G(A)$.*

For example, in the directed graph in Figure 1 the subgraph induced by the vertex set $\{1, 2, 3, 4, 5, 6\}$ has two strongly connected components, $\{1, 3, 4\}$ and $\{2, 5, 6\}$, each corresponding to a subtree of the elimination tree in Figure 5.

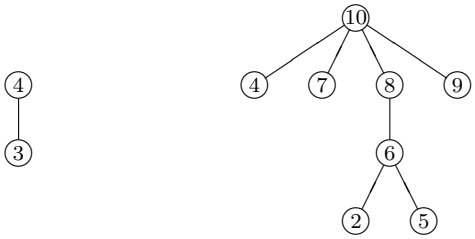
The same observation holds for the filled graph $G(A^+)$. In other words, the subtree $\mathcal{T}[k]$ is a strongly connected component of the subgraph $G_k(A^+)$ of $G(A^+)$.

As discussed at the end of section 3.1, the elimination structure $T(A)$ can be a forest. But by Corollary 3.6 each tree in the forest corresponds to a strongly connected component of $G(A)$. Thus the following result is immediate.

COROLLARY 3.7. *The directed graph $G(A)$ is strongly connected if and only if the elimination structure $T(A)$ is a tree.*

4. The elimination tree and the LU factors. We shall say that a subgraph S of the elimination tree $T(A)$ is a *pruned forest* if for every vertex $x \in S$, either the parent of x is in S or no ancestor of x is in S. In this section we characterize the row structures of L and the column structures of U in terms of pruned forests of the elimination tree.

[5] This characterization is the basis for some efficient algorithms to find the elimination tree [9]. Their complexity is $O(nm)$, where m is the number of nonzeros in A, but in practice they run at least as fast as a symbolic factorization.



Row structure L_{4*} Column structure $U_{*,10}$

FIG. 6. *The pruned forests associated with L_{4*} and $U_{*,10}$ for the matrix in Figure* 1.

Row structure L_{8*} Column structure U_{*9}

FIG. 7. *The pruned forests associated with L_{8*} and U_{*9} for the matrix in Figure* 1.

THEOREM 4.1. *The structure of each row of L and column of U is a pruned forest of $T(A)$.*

Proof. We shall prove the result for the rth row L_{r*} of L; the proof for the cth column U_{*c} of U is similar. Note that $\ell_{rx} = 0$ for $x > r$. Thus it suffices to show that $\ell_{rp} \neq 0$ if $\ell_{rx} \neq 0$ and $p = \text{FPNZ}(x) \leq r$. By Theorem 2.1, if $\ell_{rx} \neq 0$, there exists a path $r \overset{A}{\Longrightarrow}_{\min} x$. By the alternate definition of FPNZ$(*)$ in (1), there exists a cycle $p \overset{A}{\Longrightarrow}_{\max} x \overset{A}{\Longrightarrow}_{\max} p$. Combining the path with the second part of the cycle and noting that $x < p \leq r$, we obtain a path $r \overset{A}{\Longrightarrow}_{\min} p$. By Theorem 2.1 we have $\ell_{rp} \neq 0$. □

Figure 6 contains the pruned forests of row L_{4*} and column $U_{*,10}$ for the matrix in Figure 1. Both are subtrees, rooted at vertices 4 and 10, respectively.

In the symmetric case, where $U = L^T$, the row structure of L_{k*} (or, equivalently, the column structure of U_{*k}) is always a pruned subtree of the elimination tree rooted at vertex k, and the leaves of the subtree correspond to nonzeros in A (see [18]). However, this result does not hold in the unsymmetric case. For example, Figure 7 contains the pruned forests of row L_{8*} and column U_{*9} for the matrix in Figure 5. The structure of L_{8*} is $\{1, 2, 3, 4, 5, 6, 8\}$, which corresponds to two pruned subtrees of the elimination tree, one rooted at vertex 8, and the structure of U_{*9} is $\{4, 5, 6, 7, 8, 9\}$, which corresponds to four pruned subtrees, one rooted at vertex 9.

An important use of the elimination tree in the symmetric case is to capture data dependencies such as among the columns in the Cholesky factor and the frontal matrices in the multifrontal method. These relationships do not hold in the unsymmetric case.

However, the following result specifies a dependency among the diagonal entries a^+_{kk} of the factor matrix A^+. Since the result is of minor importance, we state it without proof. We shall revisit the issue of data dependency in section 6.4.

THEOREM 4.2. *The value of a^+_{qq} depends on the value of a^+_{kk} if and only if vertex q is an ancestor of vertex k in the elimination tree $T(A)$.*

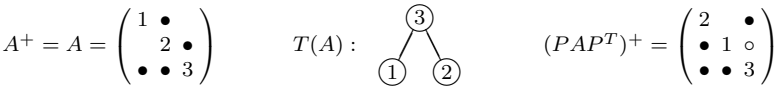


FIG. 8. *A topological ordering need not preserve the filled graph structure.*

5. Matrix reordering based on the elimination tree.

5.1. Topological orderings. Two reorderings of a matrix A are *equivalent* if their associated filled graphs are isomorphic[6] [17]. With the amount of fill-in fixed, we can then choose an equivalent reordering that has additional desirable features.

A *topological ordering* of a rooted tree numbers the children of each vertex before the vertex itself. It is well known that for symmetric matrices, any topological ordering of the elimination tree is equivalent to the original ordering and preserves the elimination tree [18].

However, for unsymmetric matrices topological orderings need not preserve the filled graph structure. For example, Figure 8 contains a 3×3 unsymmetric matrix and its elimination tree. The permutation matrix P numbers vertex 2 before vertex 1 and is therefore a topological ordering of the elimination tree. But A suffers no fill, whereas the reordered matrix PAP^T suffers one fill.

Nonetheless, topological orderings do preserve the structure of the elimination tree.

THEOREM 5.1. *Let π be a topological ordering of the elimination tree $T(A)$, and let P be the corresponding permutation matrix. The elimination tree $T(PAP^T)$ is $T(A)$ with the vertex relabeling $x \to \pi(x)$.*

Proof. The result follows directly from Theorem 3.5 and Corollary 3.4. ∎

They also preserve the pivots! For any matrix M, let $D_U(M)$ be the diagonal matrix whose diagonal elements are the same as those of the U in the LU factorization of M.

THEOREM 5.2. *Let π be a topological ordering of the elimination tree $T(A)$, and let P be the corresponding permutation matrix. Then $PD_U(A)P^T = D_U(PAP^T)$.*

Proof. For any set S of vertices in $G(A)$, let $A(S)$ denote the principal submatrix of A consisting of the rows and columns in S, and for any vertex k, let $A(k)$ denote $A(\{1, \ldots, k\})$.

Since $A = LU$ and L is unit lower triangular, we have

$$u_{kk} = \det A(k) \,/\, \det A(k-1).$$

By Corollary 3.6 the subtree $\mathcal{T}[k]$ is a strongly connected component of the subgraph $G_k(A)$ of $G(A)$. Thus there exists a permutation Q such that $QA(k)Q^T$ is block upper triangular and $A(\mathcal{T}[k])$ is one of the diagonal blocks.

Since the determinant is invariant under a symmetric permutation of the rows and columns, we can express $\det A(k)$ as the product of the determinants of these diagonal blocks. Similarly, we can express $\det A(k-1)$ as the same product with $\det A(\mathcal{T}[k])$ replaced by $\det A(\mathcal{T}[k] \setminus \{k\})$. Thus

$$u_{kk} = \det A(\mathcal{T}[k]) \,/\, \det A(\mathcal{T}[k] \setminus \{k\}).$$

The result now follows from Theorem 5.1 and the invariance of the determinant under symmetric permutations. ∎

[6] Two graphs are *isomorphic* if either can be obtained from the other by relabeling the vertices.

$$(PAP^T)^+ = \begin{pmatrix}
2 & \bullet & & \bullet & & \bullet & & \bullet \\
 & 5 & \bullet & \bullet & & & \bullet & \\
\bullet & \circ & 6 & \circ & & \circ & \circ & \circ \\
\bullet & \circ & \circ & 8 & & \bullet & \circ & \circ \\
 & & & & 9 & & & \bullet \\
 & & & & & 7 & \bullet & \\
 & & & & & & 1 & \bullet \\
 & & & & & & \bullet & 3 & \bullet \\
 & & & & \bullet & & \bullet & 4 & \bullet \\
 & \bullet & \circ & \circ & & \bullet & \circ & \circ & \circ & 10
\end{pmatrix}$$

FIG. 9. *The filled matrix for a postordering of the elimination tree of the matrix in Figure* 1.

5.2. Postorderings. Consider the subtree $\mathcal{T}[k]$ of the elimination tree rooted at vertex k, and let s_1, s_2, ..., s_t be the children of k. In a *postordering* [2] the vertices within each subtree $\mathcal{T}[s_i]$ are numbered consecutively, and vertex k is numbered immediately after the vertices in its subtrees $\mathcal{T}[s_1]$, $\mathcal{T}[s_2]$, ..., $\mathcal{T}[s_t]$.

Postorderings form an important subclass of the possible topological orderings of a given elimination tree. For symmetric matrices they are useful in such contexts as the formation of an assembly tree for the multifrontal method [6].

Figure 9 shows the reordered matrix using the postordering

$$2,\ 5,\ 6,\ 8,\ 9,\ 7,\ 1,\ 3,\ 4,\ 10$$

obtained from the elimination tree in Figure 5. Grouping together vertices belonging to the same strongly connected component of the subgraph $\mathcal{T}[10]\backslash\{10\}$ has introduced some block structure. More specifically, the vertices 2, 5, 6, and 8 in the subtree $\mathcal{T}[8]$ are numbered consecutively and form a block, as do the vertices 1, 3, and 4 in the subtree $\mathcal{T}[4]$. The number of fills is reduced from 20 to 14.

5.3. BBT postorderings. Postordering does not specify the *order* in which the subtrees are numbered. For example, in the postordering of Figure 9 the subtrees under vertex 10 are processed in the order $\mathcal{T}[8]$, $\mathcal{T}[9]$, $\mathcal{T}[7]$, $\mathcal{T}[4]$. Using a different order, such as $\mathcal{T}[4]$, $\mathcal{T}[7]$, $\mathcal{T}[9]$, $\mathcal{T}[8]$, gives rise to a different postordering and a different reordered matrix.

Consider the subtree $\mathcal{T}[k]$ of the elimination tree rooted at vertex k, and let s_1, s_2, ..., s_t be the children of k. The subgraph $\mathcal{T}[k] \setminus \{k\}$ of $G(A)$ has the vertex set

$$\mathcal{T}[k] \setminus \{k\} = \mathcal{T}[s_1] \cup \mathcal{T}[s_2] \cup \cdots \cup \mathcal{T}[s_t].$$

Form a *quotient graph* of $\mathcal{T}[k] \setminus \{k\}$ by coalescing into a single vertex the vertices in each set $\mathcal{T}[s_i]$ and coalescing into a single edge the edges (if any) from a vertex in $\mathcal{T}[s_i]$ to a vertex in $\mathcal{T}[s_j]$. By Theorem 3.5 each $\mathcal{T}[s_i]$ is a strongly connected component of the subgraph $\mathcal{T}[k] \setminus \{k\}$, so the resulting quotient graph is a dag. For example, Figure 10 shows the quotient graph with $k = 10$ for the matrix in Figure 1.

It is well known that the vertices of a dag can be arranged on a horizontal line so that all edges are directed from left to right (or from right to left). Such an ordering is called a *topological sort* [5]. For example, in the topological sort

$$\{7\},\ \{2,5,6,8\},\ \{1,3,4\},\ \{9\}$$

for the quotient graph in Figure 10, the three edges all point from left to right.

A left-to-right topological sort defines an order to process the subtrees $\mathcal{T}[s_1]$, $\mathcal{T}[s_2]$, ..., $\mathcal{T}[s_t]$ in the postordering, and the resulting submatrix is in block upper

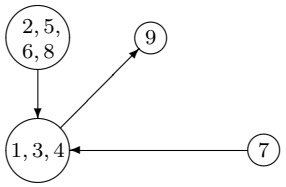


FIG. 10. *The quotient graph of* $\mathcal{T}[10] \setminus \{10\}$ *for the matrix in Figure* 1.

$(PAP^T)^+ =$

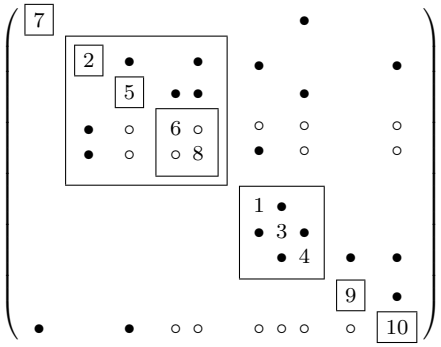


FIG. 11. *The filled matrix for an upper BBT postordering of the matrix in Figure* 1.

triangular form. When we include the subtree root k, we get a submatrix of the form

$$\begin{pmatrix} A_{11} & A_{12} & \dots & A_{1t} & A_{1k} \\ 0 & A_{22} & \dots & A_{2t} & A_{2k} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & A_{tt} & A_{tk} \\ A_{k1} & A_{k2} & \dots & A_{kt} & A_{kk} \end{pmatrix},$$

where the diagonal block A_{ii} corresponds to the subtree $\mathcal{T}[s_i]$ for $1 \le i \le t$. We shall refer to this as upper bordered block triangular (BBT) form. Note that if we use a topological sort that orders the vertices of the quotient graph from right to left, we obtain a lower BBT form that is lower bordered block triangular.

Using a topological sort to order the subtrees can be done recursively at every vertex. We shall refer to such a postordering as a *BBT postordering* and say that a matrix ordered by a BBT postordering of its elimination tree is *BBT ordered*.

Figure 11 shows the matrix in Figure 1 ordered by a BBT postordering of the elimination tree of Figure 5. The topological sort used for the subtrees under the vertex 10 is $\mathcal{T}[7]$, $\mathcal{T}[8]$, $\mathcal{T}[4]$, $\mathcal{T}[9]$, and that for the subtrees under vertex 6 is $\mathcal{T}[2]$, $\mathcal{T}[5]$. These blocks are boxed in the figure. Note that the recursive use of BBT postordering on the block $\{2, 5, 6, 8\}$ also gives a bordered block upper triangular form. In this example there are 15 fills, reduced from 20 in Figure 5.

Figure 12 shows a lower BBT postordering of the same matrix. In this case there are 13 fills. However, BBT postorderings do not always reduce the number of fills, as shown by the example in Figure 13. Nonetheless, by Theorem 5.2 they do generate the same set of pivot values as the original ordering and thus are arguably just as stable numerically.

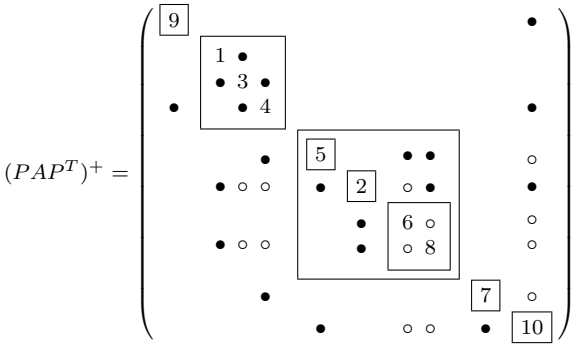


FIG. 12. *The filled matrix for a lower BBT postordering of the matrix in Figure* 1.

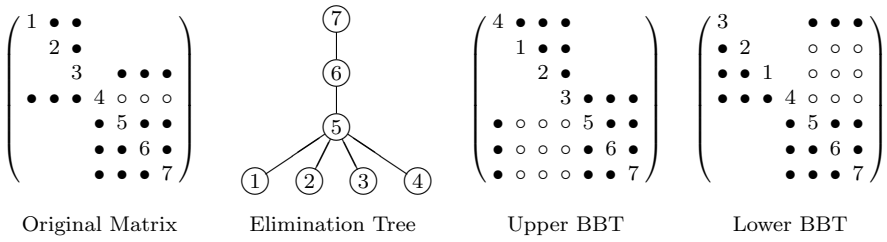


FIG. 13. *Upper and lower BBT reorderings need not lead to less fill.*

6. Properties of BBT ordered matrices.

6.1. The factor row and column structures. Theorem 4.1 characterizes the structure of the rows of L and the columns of U in terms of pruned forests of the elimination tree. The situation is simpler when the matrix is already BBT ordered.

THEOREM 6.1. *Let A be upper BBT ordered. The row structure of L_{r*} is a pruned subtree of $T(A)$ rooted at vertex r. Furthermore, the pruned subtree includes every child of r in the elimination tree.*

Proof. By Theorem 4.1 the row structure of L_{r*} is a pruned forest of the elimination tree. Assume it is not a pruned subtree rooted at r. Then there exists a vertex $j < r$ such that $\ell_{rj} \neq 0$ but j is not in the subtree $T[r]$. Thus j and r belong to different blocks with j appearing in an earlier block. This is impossible since the matrix A is upper BBT ordered.

It remains to show that $\ell_{rs} \neq 0$ for every child s of r. By the definition of FPNZ(s) there exists a cycle $r \xrightarrow{L}_{\max} s \xrightarrow{U}_{\max} r$. Assume that the path segment $r \xrightarrow{L}_{\max} s$ is not an edge, that is, that $\ell_{rs} = 0$. Then we can write that segment as $r \xrightarrow{L} i \overset{L}{\longmapsto} s$ for some vertex i with $r > i > s$ and $\ell_{is} \neq 0$. Since i is between s and r, it must belong to a different subtree of r than s, and it must belong to a later block than s. This contradicts the fact that the matrix A is already upper BBT ordered. ∎

COROLLARY 6.2. *Let A be lower BBT ordered. The column structure of U_{*c} is a pruned subtree of $T(A)$ rooted at vertex c. Furthermore, the pruned subtree includes every child of c in the elimination tree.*

Theorem 6.1 states that when A is upper BBT ordered, each nonzero in L is related to a pair of vertices, one of which is an ancestor of the other in the elimination tree. Since each row structure is a pruned subtree, not every ancestor-descendant pair of vertices is associated with a nonzero in L. However, Theorem 6.1 guarantees that

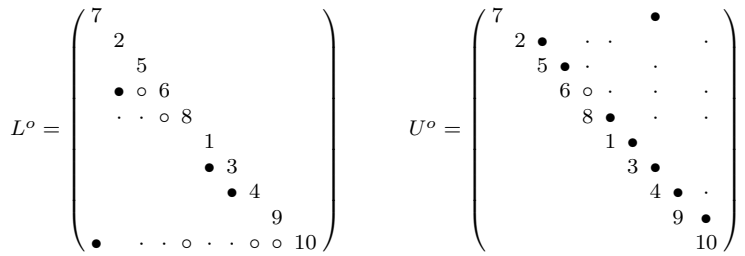


FIG. 14. *The matrix structures of the elimination dags for the matrix in Figure* 11.

such a nonzero in L can always be found for each parent-child pair in the elimination tree. By contrast the column structures of U need not be pruned subtrees. For example, four of the 10 column structures of the U in Figure 11 are pruned forests and not pruned subtrees.

6.2. The elimination dags of the filled graph. In section 2 we introduced the elimination dag $G(L^o)$ associated with the lower triangular factor L. The next result shows its connection to the elimination tree for a BBT ordered unsymmetric matrix.

THEOREM 6.3. *Let A be upper BBT ordered. The elimination dag $G(L^o)$ is the same as the elimination tree $T(A)$ when the edges in T(A) are directed from parent to child.*

Proof. By Theorem 6.1 every tree edge in $T(A)$ is in $G(L)$. The elimination dag $G(L^o)$ must contain all of these edges since none can be replaced by a path in $G(L^o)$.

Now consider a nonzero ℓ_{rc}; that is, $r > c$ and $r \overset{L}{\longmapsto} c$ is an edge in $G(L)$. By Theorem 6.1 vertex c belongs to the row subtree of L_{r*} so that vertex r is an ancestor of c. If r is not the parent of c, there is a path from r to c via edges in the elimination tree. Since all tree edges are in the elimination dag $G(L^o)$, this is also a path in $G(L^o)$, and this nontree edge in $G(L)$ will be removed by transitive reduction. Therefore the dag must be the same as the elimination tree. □

COROLLARY 6.4. *Let A be lower BBT ordered. The elimination dag $G(U^o)$ is the same as the elimination tree $T(A)$ when the edges in T(A) are directed from child to parent.*

Although the elimination dag $G(L^o)$ is the same as the elimination tree when A is upper BBT ordered, the elimination dag $G(U^o)$ need not be, as the example in Figure 14 demonstrates.

6.3. An interpretation as a depth-first tree. Depth-first search in a graph $G(M)$ starts at an initial vertex x and marks x as visited. It then searches each unvisited neighbor y of x (with edge $x \overset{M}{\longmapsto} y$) in turn, using depth-first search recursively [5].

The edges that lead to new (unmarked) vertices during a depth-first search of a strongly connected graph form a rooted tree, called a *depth-first tree*: if the edge $x \overset{M}{\longmapsto} y$ leads the search from the marked vertex x to the unmarked vertex y, then x is the parent of y in the depth-first tree.

The elimination tree for a symmetric matrix is a depth-first tree of its filled matrix [18]. In this subsection we extend this result to unsymmetric matrices.

THEOREM 6.5. *Let A be upper BBT ordered. The elimination tree $T(A)$ is a depth-first tree of the filled graph $G(A^+)$.*

Proof. Consider a depth-first search of the filled graph $G(A^+)$ subject to the following tie-breaking rule: when there is a choice of more than one vertex to explore, always select the one ordered latest in the BBT postordering. Then the search will start with the last vertex in the BBT postordering. It is easily seen that this traversal will visit the vertices of the graph in exactly the reverse order, so that the depth-first tree is the same as the elimination tree. □

For example, for the BBT ordered matrix in Figure 11 it is easy to see that[7]

Reordered:	10	9	8	7	6	5	4	3	2	1
Original:	10	9	4	3	1	8	6	5	2	7

is the sequence of vertex visits during a depth-first search subject to the tie-breaking strategy used in the proof of Theorem 6.5. This traversal gives rise to a depth-first tree that is the same as the elimination tree in Figure 5.

When the matrix is lower BBT ordered, the same result holds except that we must reverse the direction of the edges in $G(A^+)$ to account for the direction of traversal. More specifically, if A is lower BBT ordered, its transpose A^T is upper BBT ordered. But the elimination tree $T(A)$ is the same as the elimination tree $T(A^T)$, which is a depth-first tree of the filled graph of A^T.

Depth-first search can be used to classify edges in the underlying directed graph into four types [5, p. 482]:

- *tree edges*: edges in the depth-first tree;
- *forward edges*: nontree edges that connect a vertex to one of its descendants in the depth-first tree;
- *back edges*: edges that connect a vertex to one of its ancestors in the tree;
- *cross edges*: all remaining edges.

We now relate these edges with edges in the filled graph. For definiteness we assume that A is upper BBT ordered.

By Theorem 6.5 the elimination tree is a depth-first tree. By Theorem 6.1 every nonzero in the lower triangular factor L relates a vertex to one of its descendants in that tree. Therefore, in terms of the depth-first traversal, the edges in $G(L)$ are either tree edges or forward edges. Moreover, the forward edges are exactly those that are removed in forming $G(L^o)$ from $G(L)$.

On the other hand, the edges in $G(U)$ must always lead to a visited vertex since the traversal visits vertices in reverse order. Therefore these edges are either backward edges or cross edges. Note that they all point from left to right.

6.4. Data dependency in LU factorization. In the sparse Cholesky factorization of a symmetric positive definite matrix the elimination tree captures the data dependencies among the columns (or rows) of the Cholesky factor [18]. In the context of parallel factorization these dependencies can be used to find desirable orderings and choose parallel pivots.

For unsymmetric matrices the elimination tree captures data dependencies among the diagonal entries of the filled matrix (see Theorem 4.2). However, for BBT ordered unsymmetric matrices the results of Theorems 6.1 and 6.2 lead to a stronger result.

Let A be an upper BBT ordered unsymmetric sparse matrix. The rth rows of the triangular factors L and U can be obtained by applying updates from previous rows $U_{1*}, \ldots, U_{r-1,*}$. The rows used in the update are given precisely by the structure of L_{r*}.

[7] Here we provide two sequences, one using the reordered labels, the other using the originals.

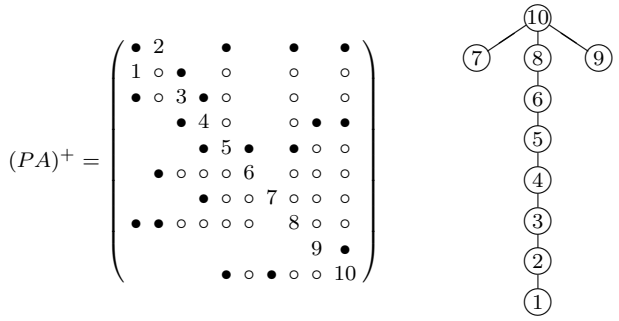


FIG. 15. *The elimination tree for the matrix in Figure* 1 *after rows* 1 *and* 2 *are exchanged.*

Thus the graph $G(L)$ captures the data dependencies among the rows of the filled matrix. Since we are only interested in the row dependency relation, we can take the transitive reduction of $G(L)$, which gives the elimination dag $G(L^o)$. By Theorem 6.3, since A is upper BBT ordered, the elimination dag $G(L^o)$ is the same as the elimination tree $T(A)$. In other words, the elimination tree captures the dependencies among the rows of the filled matrix.

The same argument can be used to show that if A is lower BBT ordered, the elimination tree provides the dependencies among the columns of the filled matrix.

7. The impact of pivoting. Thus far we have assumed that the elimination sequence is fixed. But what if a pivot is not acceptable numerically? Unrestricted row and/or column pivoting could change the entire elimination tree (see Figure 15). In this section we consider two practical forms of pivoting that have only a local impact on the tree structure.

7.1. Delayed elimination. The simplest strategy (which cannot handle all cases) is to delay the elimination of an unacceptable pivot until its value has been updated[8] and becomes acceptable [17]. For example, if vertex k is delayed until after vertex m, the sequence of rows and columns is changed from

$$1, 2, \ldots, k-1, k, k+1, \ldots, m, m+1, \ldots, n$$

to

$$1, 2, \ldots, k-1, k+1, \ldots, m, k, m+1, \ldots, n,$$

which corresponds to a symmetric reordering of the matrix. In this section we consider the effect of such a reordering.

Let $\ell \equiv k+1$ and assume that a_{kk} and $a_{\ell\ell}$ are nonzero. Let P be the permutation matrix that exchanges rows k and ℓ, i.e., that corresponds to the ordering

$$1, 2, \ldots, k-1, \ell, k, k+2, \ldots, n.$$

Let $\bar{A} \equiv PAP^t$, the matrix with vertex k delayed until after vertex ℓ. Then $\bar{A}$ also has an LU factorization, at least structurally.

The graphs $G(A)$ and $G(\bar{A})$ are isomorphic. Indeed, letting $\bar{x}$ denote the xth vertex in $G(\bar{A})$, we have that k and $\bar{\ell}$ are the same vertex, ℓ and $\bar{k}$ are the same

[8] The first update will occur when its parent is eliminated.

vertex, and x and $\bar{x}$ are the same vertex for any $x \neq k, \ell$. More importantly, the exchange has only a local effect on the elimination tree.

LEMMA 7.1. *Assume that a_{kk} and $a_{\ell\ell}$ are nonzero. Let x and y be vertices other than k and ℓ with $x < y$. There exists a cycle $y \xrightarrow{A}_{\max} x \xrightarrow{A}_{\max} y$ if and only if there exists a cycle $\bar{y} \xrightarrow{\bar{A}}_{\max} \bar{x} \xrightarrow{\bar{A}}_{\max} \bar{y}$.*

Proof. Assume that there is a cycle $y \xrightarrow{A}_{\max} x \xrightarrow{A}_{\max} y$. If it does not contain either k or ℓ, then it visits only vertices that have the same index in both graphs and thus can be written as $\bar{y} \xrightarrow{\bar{A}}_{\max} \bar{x} \xrightarrow{\bar{A}}_{\max} \bar{y}$. Otherwise we must have $y > \ell$, so that the cycle still has that form.

Since $A = P\bar{A}P^t$, the converse also holds. ∎

THEOREM 7.2. *Let vertex p be the parent of k in $T(A)$. If $p \neq \ell$, then $T(\bar{A})$ is the same as $T(A)$ up to the relabeling of vertices. If $p = \ell$, then $T(\bar{A})$ is the same except that the parent of k is now the parent q of p in $T(A)$; the parent of p is now k; and the parent of each child $s \neq k$ of k or p in $T(A)$ can now be either k or p.*

Proof. If $p \neq \ell$, then ℓ is not an ancestor of k since $\ell = k+1$. Thus the relabeling is a topological ordering of $T(A)$, and the result follows from Theorem 5.1.

Assume that $p = \ell$.

By the alternate definition of parent in (1), there is a cycle $\ell \xrightarrow{A}_{\max} k \xrightarrow{A}_{\max} \ell$, which can also be written as $\bar{k} \xrightarrow{\bar{A}}_{\max} \bar{\ell} \xrightarrow{\bar{A}}_{\max} \bar{k}$. Rearranging the two segments, we get the cycle $\bar{\ell} \xrightarrow{\bar{A}}_{\max} \bar{k} \xrightarrow{\bar{A}}_{\max} \bar{\ell}$, so that $\bar{\ell}$ (i.e., k) is the parent of $\bar{k}$ (i.e., p) in $T(\bar{A})$.

Since $\ell = p$ is a child of q in $T(A)$, by (1) there is a cycle $q \xrightarrow{A}_{\max} \ell \xrightarrow{A}_{\max} q$. Inserting the cycle $\ell \xrightarrow{A}_{\max} k \xrightarrow{A}_{\max} \ell$, we get the cycle

$$q \xrightarrow{A}_{\max} \ell \xrightarrow{A}_{\max} k \xrightarrow{A}_{\max} \ell \xrightarrow{A}_{\max} q,$$

which, since $q > \ell$, can be written as

$$\bar{q} \xrightarrow{\bar{A}}_{\max} \bar{k} \xrightarrow{\bar{A}}_{\max} \bar{\ell} \xrightarrow{\bar{A}}_{\max} \bar{k} \xrightarrow{\bar{A}}_{\max} \bar{q}$$

and as $\bar{q} \xrightarrow{\bar{A}}_{\max} \bar{\ell} \xrightarrow{\bar{A}}_{\max} \bar{q}$.

By Theorem 3.2 we have that $\bar{q}$ is an ancestor of $\bar{\ell}$ in $T(\bar{A})$. If $\bar{q}$ were not the parent of $\bar{\ell}$, there would be a cycle $\bar{x} \xrightarrow{\bar{A}}_{\max} \bar{\ell} \xrightarrow{\bar{A}}_{\max} \bar{x}$ for some $\ell < x < q$. Inserting the cycle $\bar{\ell} \xrightarrow{\bar{A}}_{\max} \bar{k} \xrightarrow{\bar{A}}_{\max} \bar{\ell}$, we get the cycle

$$\bar{x} \xrightarrow{\bar{A}}_{\max} \bar{\ell} \xrightarrow{\bar{A}}_{\max} \bar{k} \xrightarrow{\bar{A}}_{\max} \bar{\ell} \xrightarrow{\bar{A}}_{\max} \bar{x},$$

which can also be written as

$$x \xrightarrow{A}_{\max} k \xrightarrow{A}_{\max} \ell \xrightarrow{A}_{\max} k \xrightarrow{A}_{\max} x$$

since $x > \ell$. Thus we have $x \xrightarrow{A}_{\max} \ell \xrightarrow{A}_{\max} x$, which contradicts the fact that ℓ is a child of q in $T(A)$. Therefore $\bar{q}$ (i.e., q) must be the parent of $\bar{\ell}$ (i.e., k) in $T(\bar{A})$.

Consider a child s of k in $T(A)$. By (1) there is a cycle $k \xrightarrow{A}_{\max} s \xrightarrow{A}_{\max} k$, which can be written as $\bar{\ell} \xrightarrow{\bar{A}}_{\max} \bar{s} \xrightarrow{\bar{A}}_{\max} \bar{\ell}$ since $k > s$. By Theorem 3.2 we have that $\bar{\ell}$ is an ancestor of $\bar{s}$ in $T(\bar{A})$. If the parent of $\bar{s}$ in $T(\bar{A})$ were less than $\bar{k}$, then by (1) and Lemma 7.1 the parent of s in $T(A)$ would also be less than k, a contradiction. Thus the parent of $\bar{s}$ (i.e., s) must be either $\bar{\ell}$ (i.e., k) or $\bar{k}$ (i.e., p).

Consider a child vertex $t \neq k$ of ℓ (i.e., p) in $T(A)$. By (1) there is a cycle $\ell \xrightarrow{A}_{\max} t \xrightarrow{A}_{\max} \ell$. Extending it with the segments of $\ell \xrightarrow{A}_{\max} k \xrightarrow{A}_{\max} \ell$, we get the cycle

$$k \xrightarrow{A}_{\max} \ell \xrightarrow{A}_{\max} t \xrightarrow{A}_{\max} \ell \xrightarrow{A}_{\max} k,$$

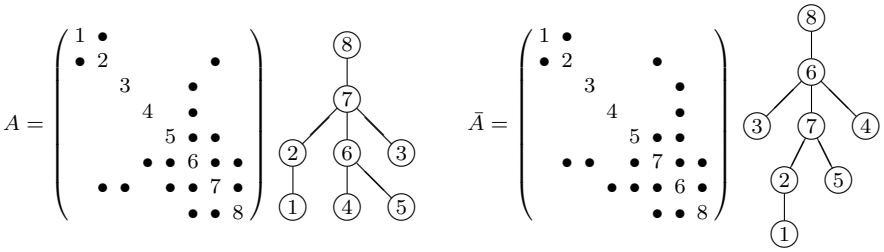


FIG. 16. *A matrix and its elimination tree before (left) and after (right) vertex* 6 *is delayed until after vertex* 7. *In the second tree each vertex* x *is labeled with the value of* $\bar{a}_{xx}$ *rather than* x.

which, since $k > t$, can be rewritten as

$$\bar{\ell} \xrightarrow{\bar{A}}_{\max} \bar{k} \xrightarrow{\bar{A}}_{\max} \bar{t} \xrightarrow{\bar{A}}_{\max} \bar{k} \xrightarrow{\bar{A}}_{\max} \bar{\ell}$$

and as $\bar{\ell} \xrightarrow{\bar{A}} \bar{t} \xrightarrow{\bar{A}}_{\max} \bar{\ell}$. Thus as before $\bar{\ell}$ must be an ancestor of $\bar{t}$ in $T(A)$, and the parent of $\bar{t}$ (i.e., t) must be either $\bar{\ell}$ (i.e., k) or $\bar{k}$ (i.e., p).

Finally, consider a vertex j that is neither k nor ℓ nor their child. By (1), Lemma 7.1, and what we have already proved, the parent of x in $T(A)$ must also be the parent of $\bar{x}$ in $T(\bar{A})$. □

Figure 16 illustrates the local effect of exchanging two vertices when the conditions in Theorem 7.2 are satisfied. Some children of k and ℓ (i.e., the vertices labeled 6 and 7) have the same parents; others do not.

For any vertex x, let $\mathcal{T}_1[x]$ be the pruned subtree of $\mathcal{T}[x]$ that includes all vertices at levels less than or equal to 1, that is, the vertex x and its children. And for any subset Y of vertices, define

$$\mathcal{T}_1[Y] = \bigcup_{y \in Y} \mathcal{T}_1[y].$$

By applying Theorem 7.2 repeatedly, we get the following result.

COROLLARY 7.3. *Assume that the elimination of vertex* k *is delayed until after the elimination of vertex* m. *Let* S *be the set consisting of* k *and its ancestors in the elimination tree* $T(A)$ *that are ordered between its old and new positions; that is,*

$$S = \{k\} \cup \{x \mid k < x \leq m \text{ and } x \text{ is an ancestor of } k\}.$$

Then the parents of vertices in $X(A) \setminus \mathcal{T}_1[S]$ *are the same in the new elimination tree.*

The set S in Corollary 7.3 is a chain of vertices in $T(A)$ starting at k. The vertices whose parents might be changed as a result of the delayed elimination are those in S and their children. Thus the structural change in the elimination tree is localized.

We use the matrix in Figure 1 to illustrate. If we delay the elimination of vertex 5 until after vertex 9, then only the parents of vertices in $\mathcal{T}_1[\{5, 6, 8\}] = \{2, 5, 6, 8\}$ may change (see Figure 17).

7.2. Off-diagonal pivoting. Delaying the elimination of vertex k until immediately after its parent p makes k the new parent of p. Thus diagonal pivoting will fail if p is also unacceptable as a pivot. On the other hand, unrestricted row and/or column pivoting can have a nonlocal effect on the elimination tree. In this section we consider a restricted form.

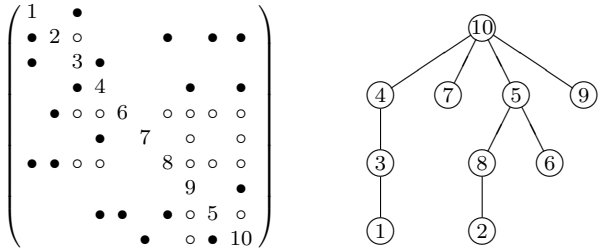


FIG. 17. *The elimination tree for the matrix in Figure* 1 *when vertex* 5 *is delayed until after vertex* 9. *Each vertex* x *is labeled with the value of* $\bar{a}_{xx}$ *rather than* x.

Let $\ell \equiv k+1$, and assume that a_{kk}, $a_{\ell k}$, $a_{k\ell}$, and $a_{\ell\ell}$ are nonzero.[9] Let P be the permutation matrix that exchanges rows k and ℓ. Then PA also has an LU factorization, at least structurally.

The difference between $G(A)$ and $G(PA)$ is local.[10] The row exchange replaces each edge $k \overset{A}{\longmapsto} v$ with $v \neq \ell$ by the edge $\ell \overset{PA}{\longmapsto} v$ and each edge $\ell \overset{A}{\longmapsto} v$ with $v \neq k$ by the edge $k \overset{PA}{\longmapsto} v$. All other edges are in both graphs. More importantly, the exchange also has a local effect on the elimination tree.

LEMMA 7.4. *Assume that* a_{kk}, $a_{k\ell}$, $a_{\ell k}$, *and* $a_{\ell\ell}$ *are nonzero. Let* x *and* y *be vertices with* $x < y$ *and* $y \neq k, \ell$. *There is a cycle* $y \overset{A}{\Longrightarrow}_{\max} x \overset{A}{\Longrightarrow}_{\max} y$ *if and only if there is a cycle* $y \overset{PA}{\Longrightarrow}_{\max} x \overset{PA}{\Longrightarrow}_{\max} y$.

Proof. Assume that there is a cycle $y \overset{A}{\Longrightarrow}_{\max} x \overset{A}{\Longrightarrow}_{\max} y$. If it does not contain either k or ℓ, then every edge belongs to both graphs so it can be written as $y \overset{PA}{\Longrightarrow}_{\max} x \overset{PA}{\Longrightarrow}_{\max} y$. Otherwise we must have $y > \ell$, and we can create such a cycle by starting with the original and replacing each edge $k \overset{A}{\longmapsto} v$ with $v \neq \ell$ by the path $k \overset{PA}{\longmapsto} \ell \overset{PA}{\longmapsto} v$ and each edge $\ell \overset{A}{\longmapsto} v$ with $v \neq k$ by the path $\ell \overset{PA}{\longmapsto} k \overset{PA}{\longmapsto} v$. (The remaining edges belong to both graphs.)

Since $A = P(PA)$, the converse also holds. ∎

THEOREM 7.5. *Assume that* a_{kk}, $a_{k\ell}$, $a_{\ell k}$, *and* $a_{\ell\ell}$ *are nonzero. Then* $T(PA)$ *is the same as* $T(A)$ *except that the parent of each child* $s \neq k$ *of* k *or* ℓ *in* $T(A)$ *can now be either* k *or* ℓ.

Proof. Since $\ell \mapsto k \mapsto \ell$ in each graph, the parent of k is ℓ in both trees.

Consider a child s of k in $T(A)$. By the alternate definition of parent in (1) there is a cycle $k \overset{A}{\Longrightarrow}_{\max} s \overset{A}{\Longrightarrow}_{\max} k$. Let $k \overset{A}{\longmapsto} v$ be the first edge in that cycle. Then $v < k$ and the remaining edges belong to both graphs. Thus we can replace that first edge by the edge $\ell \overset{PA}{\longmapsto} v$ and append the edge $k \overset{PA}{\longmapsto} \ell$ to create a cycle $\ell \overset{PA}{\Longrightarrow}_{\max} s \overset{PA}{\Longrightarrow}_{\max} \ell$.

By Theorem 3.2 vertex ℓ must be an ancestor of s in $T(PA)$. If the parent of s in $T(PA)$ were less than k, then by (1) and Lemma 7.4 the parent of s in $T(A)$ would also be less than k, a contradiction. Thus the parent of s in $T(PA)$ must be either k or ℓ.

Let $t \neq k$ be a child of ℓ in $T(A)$. By (1) there is a cycle $\ell \overset{A}{\Longrightarrow}_{\max} t \overset{A}{\Longrightarrow}_{\max} \ell$. If we replace each edge $k \overset{A}{\longmapsto} v$ in the cycle by the path $k \overset{PA}{\longmapsto} \ell \overset{PA}{\longmapsto} v$ and each edge $\ell \overset{A}{\longmapsto} v$ by the path $\ell \overset{PA}{\longmapsto} k \overset{PA}{\longmapsto} v$, we get a cycle $\ell \Rightarrow t \Rightarrow \ell$ in the subgraph $\{1, \ldots, \ell\}$ of $G(PA)$. This cycle must contain a (not necessarily proper) subcycle $\ell \overset{PA}{\Longrightarrow}_{\max} t \overset{PA}{\Longrightarrow}_{\max} \ell$. Thus, as before, k must be an ancestor of t in $T(PA)$, and the

[9] It would suffice that these values are (or may be assumed to be) structurally nonzero at the time that vertex k is eliminated.

[10] For simplicity we shall use the same symbols to represent corresponding vertices in these graphs.

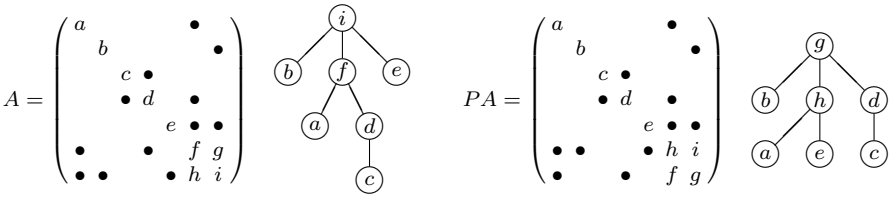


FIG. 18. *A matrix and its elimination tree before (left) and after (right) the exchange of rows* 6 *and* 7. *Each vertex* x *is labeled with the value of* $\bar{a}_{xx}$ *rather than* x.

parent of t in $T(PA)$ must be either k or ℓ.

Finally, consider a vertex x that is not a child of either k or ℓ. By (1), Lemma 7.4, and what we have proved already, if p is the parent of x in $T(A)$, then p must also be the parent of x in $T(PA)$. □

Figure 18 illustrates the local effect of exchanging two rows when the conditions in Theorem 7.5 are satisfied. Some children of k and ℓ have the same parent; others do not.

By applying Theorem 7.5 repeatedly, we get the following result.

COROLLARY 7.6. *Let* $S = \{k, k+1, \ldots, m\}$ *and assume that* $a_{ij} \neq 0$ *for* $i, j \in S$. *Let* Q *denote the permutation matrix that exchanges rows* k *and* m. *Then* $T(QA)$ *is the same as* $T(A)$ *except that the parent of each child* $s \notin S$ *in* $T(A)$ *of a vertex in* S *can now be any vertex in* S.

While this result may seem restrictive, it is sufficient to handle the kind of off-diagonal pivoting (within the fully summed rows and columns of the frontal matrix) done in an unsymmetric multifrontal code [3, 4, 13].

8. Concluding remarks. In this paper we have laid the theoretical foundation for a generalization of the elimination tree structure previously defined only for symmetric matrices. We have used this structure to find matrix reorderings that yield a bordered block triangular form and have proved some interesting properties of the reordered matrix. Finally we have shown that two restricted forms of pivoting have only a local effect on the elimination tree (although the changes can propagate upward if a pivot fails repeatedly).

We have ignored many important practical issues, such as efficient algorithms for computing or updating this tree structure and for generating BBT reorderings. We shall address them in a future paper [9], where they will be presented in the context of a new scheme for symbolic factorization.

We now discuss some potential applications of the elimination tree structure and the related BBT reorderings.

The elimination tree of a lower BBT ordered unsymmetric matrix captures the data dependencies of the factor columns (see section 6.4). This suggests a new "tree-parallel" approach to parallelizing the column-based GESP (Gaussian elimination with static pivoting) algorithm [15] that complements the blocking strategy used in `SuperLU_DIST` [16]. And since restricted pivoting has only a local effect on the elimination tree, it could be incorporated as well.

The WSMP sparse LU code [12] is an implementation of the unsymmetric multifrontal method [3, 4, 13, 14]. It uses a *data-dag* to identify independent computations and to guide the assembly of data from each frontal matrix into later ones. For BBT ordered matrices one of the elimination dags is the elimination tree (Theorem 6.3 and Corollary 6.4), so the associated data-dag will be simpler. Moreover, an augmented

elimination tree seems to offer a more refined model [10].

Finally, MUMPS [3] is another implementation of the unsymmetric multifrontal method that uses the elimination tree $T(A + A^t)$ to guide the assembly process. Again, an augmented elimination tree offers a more refined model [10].

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] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms*, Addison–Wesley, Reading, MA, 1983.

[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 and C. Puglisi, *An unsymmetrized multifrontal LU factorization*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 553–569.

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.

[6] 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.

[7] 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.

[8] 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.

[9] S. C. Eisenstat and J. W. H. Liu, *Algorithmic aspects of elimination trees for sparse unsymmetric matrices*, Technical report, Department of Computer Science, York University, Toronto, Canada, 2004.

[10] S. C. Eisenstat and J. W. H. Liu, *A tree-based dataflow model for the unsymmetric multifrontal method*, Technical report, Department of Computer Science, York University, Toronto, Canada, 2004.

[11] 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.

[12] A. Gupta, *Improved symbolic and numerical factorization algorithms for unsymmetric sparse matrices*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 529–552.

[13] A. Gupta, *Recent advances in direct methods for solving unsymmetric sparse systems of linear equations*, ACM Trans. Math. Software, 28 (2002), pp. 301–324.

[14] S. M. Hadfield and T. A. Davis, *Lost pivot recovery for an unsymmetric-pattern multifrontal method*, Technical report TR-94-029, University of Florida, 1994.

[15] X. S. Li and J. W. Demmel, *Making sparse Gaussian elimination scalable by static pivoting*, in Proceedings of the 1998 ACM/IEEE Conference on Supercomputing, San Jose, CA, IEEE Computer Society, Washington, DC, 1998.

[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, *A tree model for sparse symmetric indefinite matrix factorization*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 26–39.

[18] J. W. H. Liu, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 134–172.

[19] D. J. Rose and R. E. Tarjan, *Algorithmic aspects of vertex elimination of directed graphs*, SIAM J. Appl. Math., 34 (1978), pp. 176–197.

[20] R. Schreiber, *A new implementation of sparse Gaussian elimination*, ACM Trans. Math. Software, 8 (1982), pp. 256–276.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 3, pp. 706–734

DISPLACEMENT STRUCTURE APPROACH TO DISCRETE-TRIGONOMETRIC-TRANSFORM BASED PRECONDITIONERS OF G.STRANG TYPE AND OF T.CHAN TYPE*

THOMAS KAILATH† AND VADIM OLSHEVSKY‡

Abstract. In this paper we use a *displacement structure* approach to design a class of new preconditioners for the *conjugate gradient method* applied to the solution of large Toeplitz linear equations. Explicit formulas are suggested for the G.Strang-type and for the T.Chan-type preconditioners belonging to any of eight classes of matrices diagonalized by the corresponding discrete cosine or sine transforms. Under the standard Wiener class assumption the *clustering property* is established for all of these preconditioners, guaranteeing rapid convergence of the preconditioned conjugate gradient method. All the computations related to the new preconditioners can be done in real arithmetic, and to fully exploit this advantageous property one has to suggest a fast real-arithmetic algorithm for multiplication of a Toeplitz matrix by a vector. It turns out that the obtained formulas for the Strang-type preconditioners allow a number of representations for Toeplitz matrices leading to a wide variety of real-arithmetic multiplication algorithms based on any of eight discrete cosine or sine transforms.

Recently, transformations of Toeplitz matrices to Vandermonde-like or Cauchy-like matrices have been found to be useful in developing accurate *direct* methods for Toeplitz linear equations. In this paper we suggest further extending the range of the transformation approach by exploring it for *iterative* methods; this technique allowed us to reduce the complexity of each iteration of the preconditioned conjugate gradient method. The results of this paper were announced in [T. Kailath and V. Olshevsky, *Calcolo*, 33 (1996), pp. 191–208].

Key words. displacement structure, preconditioner, conjugate gradient, DCT, DST, discrete convolution

AMS subject classifications. 15A24, 65F10

DOI. 10.1137/S0895479896312560

1. Introduction.

1.1. Preconditioned conjugate gradient method (PCGM) for Toeplitz linear equations.

We consider the solution of a large linear system of equations $A_m x = b$ whose coefficient matrix A_m is an $m \times m$ leading submatrix of a semi-infinite real symmetric Toeplitz matrix of the form

$$A = \left[\ a_{|i-j|}\ \right] = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & \cdots \\ a_1 & a_0 & a_1 & a_2 & \ddots \\ a_2 & a_1 & a_0 & a_1 & \ddots \\ a_3 & a_2 & a_1 & a_0 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{1.1}$$

usually associated with the corresponding *generating function* $a(z) = \sum_{k=-\infty}^{\infty} a_{|k|} z^k$. Gaussian elimination ignores any special structure, thus requiring $O(m^3)$ arithmetic

*Received by the editors November 25, 1996; accepted for publication (in revised form) February 3, 2004; published electronically March 3, 2005. This work was supported in part by the NSF Awards CCR 0098222 and CCR 0242518.

http://www.siam.org/journals/simax/26-3/31256.html

†Information Systems Laboratory, Stanford University, Stanford, CA 94305 (tk@isl.stanford.com).

‡Department of Mathematics, University of Connecticut, Storrs, CT 06269 (olshevsky@math.uconn.edu, http://www.math.uconn.edu/∼olshevsky).

operations to solve $A_m x = b$. There are a number of *fast* Toeplitz solvers all taking advantage of the structure (1.1) to significantly reduce the number of operations. For example, the classical Schur and Levinson algorithms (see, e.g., [K87] and references therein) each require only $O(m^2)$ operations per system. Moreover, there are even *superfast* Toeplitz solvers with a smaller complexity of $O(m \log^2 m)$ operations. The numerical stability of such *direct* Toeplitz solvers is discussed in a number of recent papers; see, e.g., [O03] and the references therein. They reveal that although fast algorithms were believed to be unstable by their very nature, there are methods to obtain a simultaneously fast and accurate solution.

Along with considerable current efforts to develop and to stabilize direct methods, the PCGM for solving Toeplitz linear systems has garnered much attention. This is a well-known iterative procedure which computes at each iteration step two inner products of length m and one multiplication of the coefficient matrix by a vector, thus requiring $O(m \log m)$ operations per iteration. The number of iterations depends upon the clustering of the spectrum of the A_m, and if the latter is clustered around 1 having a small number of outliers, then PCGM will converge rapidly; see, e.g., [GL89].

Classical results on the eigenvalue distribution of Toeplitz matrices (see, e.g., [GS84]) indicate that we cannot expect, in general, any clustering, and that the convergence of the method will be slow. This disadvantage motivated Strang to propose the use of a certain *circulant* matrix P to reduce the number of iterations. The idea was to apply the algorithm to a *preconditioned system*

$$P^{-1}Ax = P^{-1}b, \tag{1.2}$$

where the *preconditioner* P should satisfy the following three requirements.

Property 1. The complexity of the construction of P should be small, not exceeding $O(m \log m)$ operations.

Property 2. A linear system with P should be solved in $O(m \log m)$ operations.

Property 3. The spectrum of $P^{-1}A_m$ should be clustered around 1; more precisely, the following holds:

- For any $\varepsilon > 0$ there exist integers N and s such that for any $m > N$, at most s eigenvalues of $P^{-1}A$ lie outside the interval $[1-\varepsilon, 1+\varepsilon]$.

Summarizing, if a preconditioner satisfying the above Properties 1–3 can be constructed, then the complexity of the PCGM will be reduced to only $O(m \log m)$ operations, which will be even less than the complexity of superfast direct methods.

The first (now well-known) proposed preconditioners of Strang [S86] and of T. Chan [C88] were *circulant* matrices, defined, respectively, by

$$S(A_m) = \text{ circ } (a_0, a_1, a_2, \ldots, a_2, a_1),$$

$$C(A_m) = \text{circ}\left(a_0,\ \frac{m-1}{m}a_1 + \frac{1}{m}a_{m-1},\ \frac{m-2}{m}a_2 + \frac{2}{m}a_{m-2}, \ldots, \frac{1}{m}a_{m-1} + \frac{m-1}{m}a_1\right).$$

Here circ (r) denotes a circulant matrix specified by its first row r. For these two preconditioners the first property holds by their construction, and since circulant matrices are diagonalized by the discrete Fourier transform (DFT) matrix $\mathcal{F}$, the second property is also immediately satisfied. Moreover, for the case when the generating function $a(z) = \sum_{k=-\infty}^{\infty} a_{|k|} z^k$ is a function from the Wiener class, positive on the unit circle, the third property for the Strang and T. Chan preconditioners was established in [C89], [CS89], and in [CY92], respectively.

A recent survey [CN96] gives a fairly comprehensive review of these and related results, and describes many other preconditioners, including those of R. Chan, Tyrtyshnikov, Ku and Kuo, Huckle, and others. (A thorough theoretical and numerical comparison of all different preconditioners is one of the directions of current research, indicating that the question of "which preconditioner is better" may have different answers depending upon the particular classes of Toeplitz systems and their generating functions; see, e.g., [TS96], [T95], [CN96].)

Along with their many favorable properties, circulant preconditioners unfortunately require complex arithmetic (for computing FFTs), even for real symmetric Toeplitz matrices. To overcome this disadvantage, Bini and Di Benedetto [BB90] proposed *noncirculant* analogues of the Strang and of T.Chan preconditioners, belonging to the so-called τ-class (introduced in [BC83] as the class of all matrices diagonalized by the (*real*) discrete sine I transform (DST-I) matrix). Bini and Di Benedetto established for their preconditioners Properties 1–3 under the Wiener class assumption.

In this paper we continue the work started in [S86], [C88], and [BB90] and give a systematic account of Strang-type and T.Chan-type preconditioners belonging to the classes of matrices diagonalized by other real trigonometric transforms. We consider four discrete cosine and four discrete sine transforms and refer to recent papers [OOW03], [O04] for a systematic matrix presentation of the fast algorithms for all eight transforms. For each of these eight cases we derive explicit formulas for the Strang-type and the T.Chan-type preconditioners and establish for them the above Properties 1–3 (under the standard Wiener class assumption).

This problem could perhaps be solved directly, but we have found that an interpretation in terms of *displacement structure* [KKM79] and of *partially reconstructible* matrices [KO95a] often allows us to simplify many arguments. We believe that the displacement structure approach (systematically exposed in this contribution) will be useful in addressing other problems related to preconditioning, and [CNP94] and [H95] support this anticipation.

1.2. Displacement structure approach. We next use the results of [KO95a] to briefly give an interpretation of the classical Strang and T.Chan circulant preconditioners in terms of *partially reconstructible matrices*. This technique will be further extended in the main text below. The displacement structure approach initiated by [KKM79] is based on introducing in a linear space of all $m \times m$ matrices a suitable displacement operator $\nabla(\cdot) : \mathbb{R}^{m\times m} \to \mathbb{R}^{m\times m}$ of the form

$$\nabla(R) = R - FRF^T \qquad \text{or} \qquad \nabla(R) = F^T R - RF. \tag{1.3}$$

A matrix R is said to have ∇-displacement structure, if it is mapped to a low-rank matrix $\nabla(R)$. Since a low-rank matrix can be described by a small number of parameters, a representation of a matrix by its image $\nabla(R)$ often leads to interesting results, and is useful for the design of many fast algorithms. This approach has been found to be useful for studying many different patterns of structure (for example, Toeplitz $T = [T_{i-j}]$, Vandermonde $V = [x_i^{j-1}]$, and Cauchy $C = [\frac{1}{x_i - y_j}]$) by specifying for each of them an appropriate displacement operator. For example, *Toeplitz-like matrices* are defined as having displacement structure with respect to the choice

$$\nabla_{Z_1}(R) = R - Z_1 R Z_1^T, \tag{1.4}$$

where $Z_1 =$ circ $(0, \ldots, 0, 1)$. The motivation for the above definition can be inferred from the easily verified fact that the constant-diagonal structure of a Toeplitz matrix A

in (1.1) implies that the rank of the matrix

$$\nabla_{Z_1}(A) = \left[\begin{array}{c|ccc} ? & ? & \cdots & ? \\ \hline ? & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ ? & 0 & \cdots & 0 \end{array}\right]$$

does not exceed two. If the rank of $\nabla_{Z_1}(A)$ is bigger than two but still sufficiently small, A is called Toeplitz-like. Although the latter definition of Toeplitz-like matrices has already used by several authors, it is slightly different from the standard one,

$$\nabla_{Z_0}(R) = R - Z_0 R Z_0^T,$$

where Z_0 is the lower shift matrix. The crucial difference is that ∇_{Z_1} clearly has a *nontrivial kernel*, so the image $\nabla_{Z_1}(R)$ no longer contains all the information on R. Such matrices R have been called *partially reconstructible* in [KO95a] and systematically studied there. In the Toeplitz-like case Ker ∇_{Z_1} coincides with the subspace of all circulant matrices in $\mathbb{R}^{m\times m}$, so we can observe that the Strang and T.Chan preconditioners are both chosen from Ker ∇_{Z_1}.

1.3. A proposal: ∇_{H_Q}-kernel preconditioner. The above displacement operator ∇_{Z_1} is not the only one associated with the class of Toeplitz matrices. We propose to apply the above interpretation, and develop the analogues of Strang and T.Chan preconditioners in the kernels of several other related displacement operators of the form

$$\nabla_{H_Q}(R) = H_Q^T R - R H_Q. \tag{1.5}$$

Moreover, we shall specify eight matrices H_Q for which the kernel of the corresponding displacement operator (1.5) coincides with the subspace of matrices diagonalized by any one of the eight known versions of discrete cosine/sine transforms. For each of these cases we write down the formulas for the corresponding Strang-type preconditioner and T.Chan-type preconditioner. Under the standard Wiener class assumption we establish for these new preconditioners Properties 1–3.

1.4. Fast real-arithmetic multiplication of a Toeplitz matrix by a vector. As was mentioned above, each iteration of the PCGM involves a multiplication of the coefficient matrix A and a preconditioner P by vectors. All the computations related to the new preconditioners can be done in real arithmetic. However, the standard technique for the multiplication of a Toeplitz matrix by a vector is based on the FFT, thus requiring complex arithmetic. We show that in each of the considered cases the new formulas for the Strang-type preconditioners allow us an embedding of an $m \times m$ matrix A into a larger $2m \times 2m$ matrix, which is diagonalized by the corresponding (real) discrete cosine/sine transform matrix. This observation allows us to suggest a variety of new $O(m \log m)$ real-arithmetic algorithms for the multiplication of a Toeplitz matrix by a vector, using any of eight versions of discrete cosine or sine transforms. For the DST-I case such an algorithm was suggested earlier in [BK95].

1.5. Transformations and further reduction of the cost of one iteration. Toeplitz-like matrices display just one kind of displacement structure; see (1.4). A different displacement operator

$\nabla(R) = D_x R - R D_y$ (with diagonal $D_x = \mathrm{diag}(x_1, \ldots, x_n)$ and $D_y = \mathrm{diag}(y_1, \ldots, y_n)$)

has been used to define the class of Cauchy-like matrices. Clearly, for a Cauchy matrix $C = [\frac{1}{x_i - y_j}]$ the rank of $\nabla(C) = D_x C - C D_x = [1]$ is one. Hence, if the latter rank is bigger than one but small, the matrix is called Cauchy-like.

It has been observed in many places (cf. [P90], [GO94a], [He95a], among many others) that matrices with displacement structure can be transformed from one class to another. In particular (cf. [GO94a]), the fact that $Z_1 = \mathcal{F}^* D \mathcal{F}$, where $\mathcal{F}$ is the normalized DFT matrix, allows us to transform a Toeplitz-like matrix R into a Cauchy-like matrix $\mathcal{F} A \mathcal{F}^*$. Since Cauchy-like matrices allow introducing pivoting into fast Gaussian elimination algorithms (cf. Alg. 7.1 (partial pivoting) and Alg. 6.1 (symmetric pivoting) of [GO94b]), this idea has been found to be useful to numerically reliable *direct* methods for solving Toeplitz linear equations (see, e.g., [He95a]); for the first accurate algorithms of this kind see [GKO95], [KO95a], as well as [KO95b], [KO94], [BKO94], [SB95], [Gu98].

In this paper we suggest exploiting the transformation technique for *iterative* methods and replacing a preconditioned system $(P^{-1}A)x = P^{-1}b$ by a transformed system

$$(\mathcal{F} P^{-1} \mathcal{F}^*)(\mathcal{F} A \mathcal{F}^*)(\mathcal{F} x) = (\mathcal{F} P^{-1}) b. \tag{1.6}$$

Two advantages of the latter equation are that (a) the transformed preconditioner $\mathcal{F} P^{-1} \mathcal{F}^*$ is a diagonal matrix, so the cost of computing a matrix vector product for it is just linear and that (b) it can be shown that a Cauchy-like matrix $\mathcal{F} A \mathcal{F}^*$ can be multiplied by a vector with exactly the same complexity as for the initial Toeplitz matrix A. Hence (1.6) allows us to use only four FFTs at each iteration (cf. [H94]).

In this paper we propose the exact counterpart

$$(\mathcal{T} P^{-1} \mathcal{T}^T)(\mathcal{T} A \mathcal{T}^T)(\mathcal{T} x) = (\mathcal{T} P^{-1}) b \tag{1.7}$$

of such a technique for all eight discrete cosine/sine transforms $\mathcal{T}$. Again, in all sixteen considered cases (a) each of the transformed preconditioners $\mathcal{T} P^{-1} \mathcal{T}^*$ is a diagonal matrix and (b) a Cauchy-like matrix $\mathcal{T} A \mathcal{T}^T$ can be multiplied by a vector with exactly the same complexity as for the initial Toeplitz matrix A, i.e., four cosine or sine transforms. So, all preconditioners, while dramatically reducing the number of iterations, do not increase the cost of a single iteration.

2. Partially reconstructible matrices. We shall address the problem of constructing discrete-transform based preconditioners in the second part of the paper, and start here with necessary definitions and related facts on displacement structure and partially reconstructible matrices. Let us consider a displacement operator

$$\nabla_{\{F,A\}}(R) = F \cdot R - R \cdot A \tag{2.1}$$

and recall the following standard definitions.

- A number $\alpha = \mathrm{rank} \nabla_{\{F,A\}}(R)$ is called the $\nabla_{\{F,A\}}$-*displacement rank* of R. (A matrix R is said to have a $\nabla_{\{F,A\}}$-*displacement structure* if α is small compared to the size of R.)
- A pair of rectangular $n \times \alpha$ matrices $\{G, B\}$ in any possible factorization

  $$\nabla_{\{F,A\}}(R) = F \cdot R - R \cdot A = G \cdot B^T \tag{2.2}$$

  is called a $\nabla_{\{F,A\}}$-*generator* of R.

If the matrices F and A have no common eigenvalues, $\nabla_{\{F,A\}}$ is invertible, so its generator contains a complete information on R. For our purposes in this paper it will be necessary to consider another case where the displacement operator

$$\nabla_F(R) = F^T \cdot R - R \cdot F \tag{2.3}$$

clearly has a nontrivial kernel. Such R have been called *partially reconstructible* in [KO95a], because now only part of the information on R is contained in $\{G, B\}$. Following [KO95a] we shall refer to a triple $\{G, J, R_{\mathcal{K}}\}$ as a ∇_F-generator of R, where the latter three matrices are defined as follows.

- Since $\nabla_F(R^T) = -\nabla_F(R)$, we can write

$$\nabla_F(R) = F^T \cdot R - R \cdot F = G \cdot J \cdot G^T, \quad \text{with} \quad J^T = -J \in \mathbb{R}^{\alpha\times\alpha}. \tag{2.4}$$

- Further, let us decompose

$$R = R_{\mathcal{K}} + R_{\mathcal{K}^\perp} \quad \text{with respect to} \quad \mathbb{R}^{n\times n} = \mathcal{K} \oplus \mathcal{K}^\perp, \tag{2.5}$$

where $\mathcal{K} =$ Ker ∇_F and the orthogonality in $\mathbb{R}^{n\times n}$ is defined using the inner product

$$\langle A, B\rangle = \mathrm{tr}(B^* \cdot A), \qquad A, B \in \mathbb{R}^{n\times n}, \tag{2.6}$$

with $\mathrm{tr}(A)$ denoting the sum of all diagonal entries of A, or, equivalently, the sum of eigenvalues of A. Note that the latter inner product induces the Frobenius norm in $\mathbb{R}^{n\times n}$.

Clearly, now all the information on R is contained in the newly defined generator, $\{G, J, R_{\mathcal{K}}\}$.

3. Polynomial Hankel-like matrices.

3.1. Polynomial Hankel-like matrices. In this paper we exploit a special displacement operator

$$\nabla_{H_Q}(R) = H_Q^T \cdot R - R \cdot H_Q = GJG^T, \tag{3.1}$$

with the upper *Hessenberg* matrix

$$H_Q = \begin{bmatrix} a_{01} & a_{02} & \cdots & \cdots & a_{0,n} \\ a_{11} & a_{12} & \cdots & \cdots & a_{1,n} \\ 0 & a_{22} & \cdots & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{n-1,n-1} & a_{n-1,n} \end{bmatrix}. \tag{3.2}$$

The latter has been called in [MB79] a *confederate* matrix of the associated system of polynomials $Q = \{Q_0(x), Q_1(x), \ldots, Q_n(x)\}$ defined by

$$x \cdot Q_{k-1}(x) = a_{k,k} \cdot Q_k(x) + a_{k-1,k} \cdot Q_{k-1}(x) + \cdots + a_{0,k} \cdot Q_0(x). \tag{3.3}$$

We shall refer to matrices having low ∇_{H_Q}-displacement rank as *polynomial Hankel-like matrices*; an explanation for this nomenclature will be offered in section 3.4 after presenting the following example.

3.2. Example. Classical Hankel and Hankel-like matrices. For the simplest polynomial system $P = \{1, x, x^2, \dots, x^{n-1}, Q_n(x)\}$, its confederate matrix trivially reduces to the companion matrix

$$H_P = \begin{bmatrix} 0 & 0 & \cdots & 0 & -\frac{q_0}{q_n} \\ 1 & 0 & \cdots & 0 & -\frac{q_1}{q_n} \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & -\frac{q_{n-1}}{q_n} \end{bmatrix}$$

of $Q_n(x) = q_n x^n + \cdots + q_1 x + q_0$. Now, it is straightforward to check that the shift-invariance-property of a Hankel matrix, $R = \begin{bmatrix} h_{i+j} \end{bmatrix}_{0 \le i,j \le n-1}$, implies that

$$\nabla_{H_P}(R) = H_P^T R - R H_P = \begin{bmatrix} e_n & g \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} e_n & g \end{bmatrix}^T, \tag{3.4}$$

where e_n is the last coordinate vector and

$$g = \begin{bmatrix} h_{n-2} \\ \vdots \\ h_{2n-2} \\ 0 \end{bmatrix} + \frac{1}{q_n} H \begin{bmatrix} q_0 \\ \vdots \\ q_{n-2} \\ q_{n-1} \end{bmatrix}.$$

Briefly, the ∇_{H_P}-displacement rank of an arbitrary Hankel matrix does not exceed two. Hence matrices with small ∇_{H_P}-displacement rank (not just $\alpha \le 2$) are referred to as *Hankel-like* matrices.

3.3. Diagonalization of confederate matrices. To explain the name polynomial Hankel-like matrices we shall need the following result, which will be widely used in what follows. It can be easily checked by direct multiplication (cf. [MB79]) that the confederate matrix is diagonalized by the *polynomial Vandermonde matrix* V_Q,

$$H_Q = V_Q^{-1} D_x V_Q, \tag{3.5}$$

where

(3.6)

$$V_Q = \begin{bmatrix} Q_0(x_1) & Q_1(x_1) & \cdots & Q_{n-1}(x_1) \\ Q_0(x_2) & Q_1(x_2) & \cdots & Q_{n-1}(x_2) \\ \vdots & \vdots & & \vdots \\ Q_0(x_n) & Q_1(x_n) & \cdots & Q_{n-1}(x_n) \end{bmatrix}, \qquad D_x = \operatorname{diag}(x_1, x_2, \dots, x_n).$$

Here $\{x_k\}$ are the zeros of $Q_n(x)$, which in our application here always will be n distinct numbers, so we shall impose this restriction throughout the paper.

3.4. Change of basis. Since $H_P = V_P^{-1} D_x V_P$ (see, e.g., (3.5)), we have

$$H_Q = S_{PQ}^{-1} H_P S_{PQ} \qquad \text{with} \qquad S_{PQ} = V_P^{-1} V_Q. \tag{3.7}$$

Now, using (3.7) and (3.4) one sees that the ∇_{H_Q}-displacement rank of $S_{PQ}^T R S_{PQ}$ (with Hankel R) also does not exceed two. We refer to such $S_{PQ}^T R S_{PQ}$ as *polynomial*

Hankel matrices (or Hankel matrices represented in the polynomial basis Q), because the similarity matrix $S_{PQ} = [s_{i,j}]_{1\le i,j\le n}$ can be easily shown to be an upper triangular matrix with the entries being the coefficients of $Q_k(x) = \sum_{i=0}^{k} s_{i+1,k+1}x^k$. Therefore the more general matrices having low ∇_{H_Q}-displacement rank (not just $\alpha \le 2$) are called *polynomial Hankel-like* matrices.

Since (3.1) has a nontrivial kernel, such R are partially reconstructible. Our next goal is to describe the kernel of ∇_{H_Q}.

4. Transformation to Cauchy-like matrices and the kernel of $\boldsymbol{\nabla_{H_Q}}$. Recently, transformation of structured matrices from one class to another has been found to be useful to design for them many efficient algorithms. In this paper we exploit an approach of [KO95a] for transformation of partially reconstructible matrices to transform polynomial Hankel-like matrices into Cauchy-like matrices, defined as having low displacement rank with respect to the simplest displacement operator

$$\nabla_{D_x}(R) = D_x R - R D_x \tag{4.1}$$

with a diagonal matrix D_x. In fact, (3.5) immediately implies the following statement.

PROPOSITION 4.1. *Let R be a polynomial Hankel-like matrix in* (3.1)*, given by its ∇_{H_Q}-generator $\{G, J, R_{\mathcal{K}}\}$, and let W_Q denote an arbitrary invertible diagonal matrix. Then $W_Q^{-T}V_Q^{-T}RV_Q^{-1}W_Q^{-1}$ is a Cauchy-like matrix with a ∇_{D_x}-generator*

$$\{W_Q^{-T}V_Q^{-T}G,\ \ J,\ \ W_Q^{-T}V_Q^{-T}R_{\mathcal{K}}V_Q^{-1}W_Q^{-1}\}. \tag{4.2}$$

Since the kernel of ∇_{D_x} is easy to describe (it is the subspace of all diagonal matrices), the above proposition implies the next statement.

PROPOSITION 4.2. *Let H_Q, V_Q and D_x be defined by* (3.2) *and* (3.6)*, where we assume that the diagonal entries of D_x are m different numbers. The kernel of $\nabla_{H_Q}(\cdot)$ in* (3.1) *has the form*

$$\mathcal{K} = \operatorname{span}\{(H_Q^T)^k \cdot (V_Q^T W_Q^2 V_Q),\qquad k = 0, 1, \ldots, n-1\}, \tag{4.3}$$

where W_Q is an arbitrary invertible diagonal matrix.

Finally, by replacing in (4.3) powers $(H_Q^T)^k$ by $Q_k(H_Q^T)$, and using (4.2) we obtain the following statement.

COROLLARY 4.3. *A matrix $R \in \mathcal{K} =$ Ker ∇_{H_Q} given by*

$$R = \sum_{k=0}^{n-1} r_k \cdot Q_k(H_Q^T) \cdot (V_Q^T W_Q^2 V_Q)$$

can be diagonalized as follows:

$$W_Q^{-T}V_Q^{-T}RV_Q^{-1}W_Q^{-1} = \begin{bmatrix} r(x_1) & & \\ & \ddots & \\ & & r(x_n) \end{bmatrix},$$

where the diagonal entries are computed via a polynomial Vandermonde transform

$$\begin{bmatrix} r(x_1) \\ \vdots \\ r(x_n) \end{bmatrix} = V_Q \begin{bmatrix} r_0 \\ \vdots \\ r_{n-1} \end{bmatrix}.$$

TABLE 1
Discrete trigonometric transforms.

	Discrete transform	Inverse transform
DCT-I	$C_N^I = \sqrt{\frac{2}{N-1}} \left[\ \eta_k \eta_{N-1-k} \eta_j \eta_{N-1-j} \cos \frac{kj\pi}{N-1} \ \right]_{k,j=0}^{N-1}$	$[C_N^I]^{-1} = [C_N^I]^T = C_N^I$
DCT-II	$C_N^{II} = \sqrt{\frac{2}{N}} \left[\ \eta_k \cos \frac{k(2j+1)\pi}{2N} \ \right]_{k,j=0}^{N-1}$	$[C_N^{II}]^{-1} = [C_N^{II}]^T = C_N^{III}$
DCT-III	$C_N^{III} = \sqrt{\frac{2}{N}} \left[\ \eta_j \cos \frac{(2k+1)j\pi}{2N} \ \right]_{k,j=0}^{N-1}$	$[C_N^{III}]^{-1} = [C_N^{III}]^T = C_N^{II}$
DCT-IV	$C_N^{IV} = \sqrt{\frac{2}{N}} \left[\ \cos \frac{(2k+1)(2j+1)\pi}{4N} \ \right]_{k,j=0}^{N-1}$	$[C_N^{IV}]^{-1} = [C_N^{IV}]^T = C_N^{IV}$
DST-I	$S_N^I = \sqrt{\frac{2}{N+1}} \left[\ \sin \frac{kj}{N+1}\pi \ \right]_{k,j=1}^{N}$	$[S_N^I]^{-1} = [S_N^I]^T = S_N^I$
DST-II	$S_N^{II} = \sqrt{\frac{2}{N}} \left[\ \eta_k \sin \frac{k(2j-1)}{2N}\pi \ \right]_{k,j=1}^{N}$	$[S_N^{II}]^{-1} = [S_N^{II}]^T = S_N^{III}$
DST-III	$S_N^{III} = \sqrt{\frac{2}{N}} \left[\ \eta_j \sin \frac{(2k-1)j}{2N}\pi \ \right]_{k,j=1}^{N}$	$[S_N^{III}]^{-1} = [S_N^{III}]^T = S_N^{II}$
DST-IV	$S_N^{IV} = \sqrt{\frac{2}{N}} \left[\ \sin \frac{(2k-1)(2j-1)}{4N}\pi \ \right]_{k,j=1}^{N}$	$[S_N^{IV}]^{-1} = [S_N^{IV}]^T = S_N^{IV}$

Here we may note that the idea of displacement is to replace operations on the n^2 entries of an $n \times n$ structured matrix by manipulation on a smaller number $O(n)$ of parameters. The results of sections 3 and 4 are based on the displacement equation (3.1), which describes R by the entries of H_Q and $\{G, J, R_{\mathcal{K}}\}$. In the general situation matrix H_Q itself involves $O(n^2)$ parameters, so such a representation is no longer efficient. In the next section we specify the results of sections 3 and 4, and list eight cases for which the above approach is beneficial.

5. Orthonormal polynomials and discrete trigonometric transforms.

5.1. Orthonormal polynomials. Examination of the propositions in the previous section indicates that the kernel of ∇_{H_Q} will have the simplest form in the case when there is a diagonal matrix W_Q such that the matrix $T_Q = W_Q V_Q$ is orthonormal; see, e.g., (4.3). It is easy to see that the latter condition is satisfied when the polynomials in $\{Q_k(x)\}$ are orthonormal with respect to the discrete inner product

$$\langle p(x), q(x) \rangle = \sum_{k=1}^{n} p(x_k) q(x_k) w_k^2,$$

where the nodes $\{x_k\}$ are the zeros of $Q_n(x)$, and the weights w_k are diagonal entries of W_Q. Moreover, in this case the polynomials $\{Q_k(x)\}$ satisfy three-term recurrence relations so their confederate matrix reduces to the corresponding Jacobi (i.e., symmetric tridiagonal) matrix.

5.2. Discrete cosine and sine transforms. Recall that our aim in this paper is to construct preconditioners diagonalized by discrete cosine or sine transform matrices, formally defined in Table 1, where

$$\eta_k = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0, N, \\ 1 & \text{otherwise.} \end{cases}$$

TABLE 2
First n polynomials.

	$\{Q_0,$	Q_1	$\ldots,$	Q_{n-2}	$Q_{n-1}\}$
DCT-I	$\{\frac{1}{\sqrt{2}}T_0,$	$T_1,$	$\ldots,$	$T_{n-2},$	$\frac{1}{\sqrt{2}}T_{n-1}\}$
DCT-II	$\{U_0,$	$U_1-U_0,$	$\ldots,$		$U_{n-1}-U_{n-2}\}$
DCT-III	$\{\frac{1}{\sqrt{2}}T_0,$	$T_1,$	$\ldots,$		$T_{n-1}\}$
DCT-IV	$\{U_0,$	$U_1-U_0,$	$\ldots,$		$U_{n-1}-U_{n-2}\}$
DST-I	$\{U_0,$	$U_1,$	$\ldots,$		$U_{n-1}\}$
DST-II	$\{U_0,$	$U_1+U_0,$	$\ldots,$		$U_{n-1}+U_{n-2}\}$
DST-III	$\{U_0,$	$U_1,$	$\ldots,$	$U_{n-2},$	$\frac{1}{\sqrt{2}}U_{n-1}\}$
DST-IV	$\{U_0,$	$U_1+U_0,$	$\ldots,$		$U_{n-1}+U_{n-2}\}$

TABLE 2 (*cont.*)
The last polynomial $Q_n(x)$.

	Q_n	zeros of Q_n
DCT-I	$xT_{n-1}-T_{n-2}$	$\{\cos(\frac{k\pi}{N-1})\}_0^{N-1}$
DCT-II	$U_n-2U_{n-1}+U_{n-2}$	$\{\cos(\frac{k\pi}{N})\}_0^{N-1}$
DCT-III	T_n	$\{\cos(\frac{(2k+1)\pi}{2N})\}_)^{N-1}$
DCT-IV	$2T_n$	$\{\cos(\frac{(2k+1)\pi}{2N})\}_)^{N-1}$
DST-I	U_n	$\{\cos(\frac{k\pi}{N+1})\}_1^{N}$
DST-II	$U_n+2U_{n-1}+U_{n-2}$	$\{\cos(\frac{k\pi}{N})\}_1^{N}$
DST-III	T_n	$\{\cos(\frac{(2k-1)\pi}{2N})\}_1^{N}$
DST-IV	$2T_n$	$\cos(\frac{(2k-1)\pi}{2N})\}_1^{N}$

Note that we use a slightly different definition for the discrete cosine I transform (DCT-I), ensuring that now all the discrete transform matrices in Table 1 are orthogonal. The modified DCT-I can be transformed into the regular one by just appropriate scaling.

Now, using the fact that the Chebyshev polynomials of the first and second kind,

$$T_k(x)=\cos(k\arccos x),\qquad U_k(x)=\frac{\sin((k+1)\arccos x)}{\sin(\arccos x)}, \tag{5.1}$$

are essentially cosines and sines, we obtain that all the discrete transform matrices T_Q in Table 1 can be seen as the orthogonal matrices $W_Q\cdot V_Q$ defined by orthonormal (Chebyshev-like) polynomials $\{Q_k(x)\}_{k=0}^{n-1}\}$ specified in Table 2, and the weight matrices W_Q specified in Table 3. We therefore adopt the designation

$$T_Q=W_QV_Q$$

for all eight transform matrices in Table 1 by associating them with the corresponding polynomial systems Q.

For each of the eight systems $\{Q_k(x)\}_{k=0}^{n}$ the first part of Table 2 lists the first n polynomials. To specify V_Q we also have to define the nodes $\{x_k\}_{k=1}^{n}$, or, equivalently, the last polynomial $Q_n(x)$, which is done in the second part of Table 2.

TABLE 3
The diagonal matrices W_Q.

DCT-I	$C_N^I = W_Q \cdot V_Q$	with	$W_Q = \sqrt{\frac{2}{N-1}}\mathrm{diag}(\frac{1}{\sqrt{2}}, 1, \ldots, 1, \frac{1}{\sqrt{2}})$
DCT-II	$C_N^{II} = W_Q \cdot V_Q$	with	$W_Q = \sqrt{\frac{2}{N}}\mathrm{diag}(\frac{1}{\sqrt{2}}, \cos(\frac{\pi}{2N}), \ldots, \cos(\frac{(N-1)\pi}{2N}))$
DCT-III	$C_N^{III} = W_Q \cdot V_Q$	with	$W_Q = \sqrt{\frac{2}{N}} \cdot I$
DCT-IV	$C_N^{IV} = W_Q \cdot V_Q$	with	$W_Q = \sqrt{\frac{2}{N}}\mathrm{diag}(\cos(\frac{\pi}{4N}), \cos(3\frac{\pi}{4N}) \ldots, \cos(\frac{(2N-1)\pi}{4N}))$
DST-I	$S_N^I = W_Q \cdot V_Q$	with	$W_Q = \sqrt{\frac{2}{N+1}}\mathrm{diag}(\sin(\frac{\pi}{N+1}), \ldots, \sin(\frac{N\pi}{N+1}))$
DST-II	$S_N^{II} = W_Q \cdot V_Q$	with	$W_Q = \sqrt{\frac{2}{N}}\mathrm{diag}(\sin(\frac{\pi}{2N}), \ldots, \sin(\frac{(N-1)\pi}{2N}), \frac{1}{\sqrt{2}}\sin(\frac{\pi}{2}))$
DST-III	$S_N^{III} = W_Q \cdot V_Q$	with	$W_Q = \sqrt{\frac{2}{N}}\mathrm{diag}(\sin(\frac{\pi}{2N}), \sin(\frac{3\pi}{2N}), \ldots, \frac{1}{\sqrt{2}}\sin(\frac{(2N-1)\pi}{2N}))$
DST-IV	$S_N^{IV} = W_Q \cdot V_Q$	with	$W_Q = \sqrt{\frac{2}{N}}\mathrm{diag}(\sin(\frac{\pi}{4N}), \sin(\frac{3\pi}{4N}), \ldots, \sin(\frac{(2N-1)\pi}{4N}))$

Finally, we specify the corresponding confederate matrices H_Q in Table 4.

All the proofs are straightforward and based on the well-known recurrence relations

$$T_0(x) = 1, \qquad T_1 = xT_0(x), \qquad T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x),$$

$$U_0(x) = 1, \qquad U_1 = 2xU_0(x), \qquad U_k(x) = 2xU_{k-1}(x) - U_{k-2}(x).$$

In the second part of the paper we shall use the eight displacement operators of the form

$$\nabla_{H_Q}(R) = H_Q R - R H_Q$$

with eight Jacobi matrices listed in Table 4 to design real-arithmetic discrete cosine/sine transform based preconditioners. The expressions for the preconditioners will be obtained by using certain auxiliary formulas, which are presented in the next section.

6. Strang-type preconditioners.

6.1. Definition of Strang-type preconditioners. Now we apply the technique developed in the first part of the paper to design a family of preconditioners for Toeplitz matrices. Consider a generating function of the form

$$a(x) = \sum_{k=-\infty}^{\infty} a_k z^k, \qquad a_k = a_{-k} \in \mathbb{R},$$

which we assume (a) to be from the Wiener class, i.e.,

$$\sum_{k=-\infty}^{\infty} |a_k| \leq \infty,$$

and (b) to have positive values on the unit circle

$$f(z) > 0, \qquad |z| = 1.$$

TABLE 4
The confederate matrices H_Q.

DCT-I	$H_Q = \text{tridiag}\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 \cdots 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{\sqrt{2}} \end{bmatrix}$
DCT-II	$H_Q = \text{tridiag}\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 \cdots 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$
DCT-III	$H_Q = \text{tridiag}\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 \cdots 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$
DCT-IV	$H_Q = \text{tridiag}\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 \cdots 0 & 0 & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$
DST-I	$H_Q = \text{tridiag}\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 \cdots 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$
DST-II	$H_Q = \text{tridiag}\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & 0 & 0 \cdots 0 & 0 & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$
DST-III	$H_Q = \text{tridiag}\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & \sqrt{\frac{1}{2}} \\ 0 & 0 & 0 \cdots 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & \sqrt{\frac{1}{2}} \end{bmatrix}$
DST-IV	$H_Q = \text{tridiag}\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & 0 & 0 \cdots 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$

As is well known, these two conditions guarantee that all leading submatrices A_m $(m = 1, 2, \ldots)$ of the associated infinite real symmetric Toeplitz matrix

$$A = \left[\ a_{|i-j|}\ \right] = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & \cdots \\ a_1 & a_0 & a_1 & a_2 & \ddots \\ a_2 & a_1 & a_0 & a_1 & \ddots \\ a_3 & a_2 & a_1 & a_0 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

are positive definite: $A_m > 0$. Our next goal is to construct for A_m a good preconditioner $S_Q(A_m)$ from the class $\mathcal{K}_Q = \text{Ker}\ \nabla_{H_Q}$, where H_Q is one of the eight matrices in Table 4. The preconditioners are defined by

$$S_Q(A) = \sum_{j=0}^{m-1} r_j \cdot Q_j(H_Q), \tag{6.1}$$

where the coefficients $\{r_k\}$ are listed in Table 5.

Note that the DST-I-based preconditioner was designed earlier in [BB90] and was also discussed in [BK95], [H95].

TABLE 5
Definition of the Strang-type preconditioners. The coefficients of the decomposition (6.1) *of* $S_Q(A)$.

	r_0	r_1	r_2	$\cdots$	r_{m-3}	r_{m-2}	r_{m-1}
DCT-I	$\sqrt{2}\cdot a_0$	$2a_1$	$2a_2$	$\cdots$	$2a_{m-3}$	$2a_{m-2}$	$2\sqrt{2}\cdot a_{m-1}$
DCT-II	a_0+a_1	a_1+a_2	a_2+a_3	$\cdots$	$a_{m-3}+a_{m-2}$	$a_{m-2}+a_{m-1}$	a_{m-1}
DCT-III	$\sqrt{2}\cdot a_0$	$2a_1$	$2a_2$	$\cdots$	$2a_{m-3}$	$2a_{m-2}$	$2a_{m-1}$
DCT-IV	a_0+a_1	a_1+a_2	a_2+a_3	$\cdots$	$a_{m-3}+a_{m-2}$	$a_{m-2}+a_{m-1}$	a_{m-1}
DST-I	a_0-a_2	a_1-a_3	a_2-a_4	$\cdots$	$a_{m-3}-a_{m-1}$	a_{m-2}	a_{m-1}
DST-II	a_0-a_1	a_1-a_2	a_2-a_3	$\cdots$	$a_{m-3}-a_{m-2}$	$a_{m-2}-a_{m-1}$	a_{m-1}
DST-III	a_0-a_2	a_1-a_3	a_2-a_4	$\cdots$	$a_{m-3}-a_{m-1}$	a_{m-2}	$\sqrt{2}a_{m-1}$
DST-IV	a_0-a_1	a_1-a_2	a_2-a_3	$\cdots$	$a_{m-3}-a_{m-2}$	$a_{m-2}-a_{m-1}$	a_{m-1}

6.2. Properties of the Strang-type preconditioners. The motivation behind the definitions of Table 5 is that they imply several properties (shared with the classical Strang preconditioner) that are listed next.

Property 4. For any $\varepsilon>0$ there exist $M>0$ so that for $m>M$ the spectrum of $S_Q(A_m)$ lies in the interval $[\min_{|z|=1}a(z)-\varepsilon, \max_{|z|=1}a(z)+\varepsilon]$.

Property 5. All Strang-type preconditioners $S_Q(A)$ defined in Table 5 are positive definite matrices for sufficiently large m.

Property 6. For all Strang-type preconditioners in Table 5 we have that $\|S_Q(A)\|_2$ and $\|S_Q(A)^{-1}\|_2$ are uniformly bounded independently of m.

Properties 5 and 6 are easily deduced from Property 4, so we need to establish only the latter. In order to prove that the Strang-type preconditioners all have Property 4 we need to specify Corollary 4.3 for each of the eight cases and obtain the description of the spectrum of $S_Q(A_m)$. This is done in the next statement.

COROLLARY 6.1. *Let* H_Q *be one of the matrices in Table* 4, $\mathcal{K}_Q = Ker\,\nabla_{H_Q}$, *and let*

$$R=\sum_{k=0}^{n-1} r_k Q_k(H_Q) \tag{6.2}$$

be a decomposition of $R\in\mathcal{K}_Q$ *with respect to a basis*

$$\{Q_k(H_Q), k=0,1,\ldots,n-1\} \tag{6.3}$$

in $\mathcal{K}_Q$. *Then*

$$T_Q R T_Q^T = \mathrm{diag}(\lambda_1,\ldots,\lambda_n), \tag{6.4}$$

where T_Q *is the corresponding discrete transform from Table* 1, *and*

$$\begin{bmatrix}\lambda_1\\ \vdots\\ \lambda_n\end{bmatrix} = W_Q^{-1}T_Q\begin{bmatrix}r_0\\ \vdots\\ r_{n-1}\end{bmatrix}, \tag{6.5}$$

where W_Q *is the corresponding weight matrix from Table* 3.

Specifying the latter corollary to each of the eight cases, we obtain that the eigenvalues of $S_Q(A_m)$ will have the form shown in Table 6. (In Table 6 we list

TABLE 6
Eigenvalues $\{\lambda_1, \ldots, \lambda_m\}$ *of* $S_Q(A)$.

DCT-I	$\lambda_k = a_0 + 2\sum_{j=1}^{m-1} a_j \cdot \cos\frac{(k-1)j\pi}{m-1}$	=	$a_m(z_k)$	where	$z_k = e^{\frac{k-1}{m-1}\pi i}$
DCT-II	$\lambda_k = a_0 + 2\sum_{j=1}^{m-1} a_j \cdot \cos\frac{(k-1)j\pi}{m}$	=	$a_m(z_k)$	where	$z_k = e^{\frac{k-1}{m}\pi i}$
DCT-III	$\lambda_k = a_0 + 2\sum_{j=1}^{m-1} a_j \cdot \cos\frac{(2k-1)j\pi}{2m}$	=	$a_m(z_k)$	where	$z_k = e^{\frac{2k-1}{2m}\pi i}$
DCT-IV	$\lambda_k = a_0 + 2\sum_{j=1}^{m-1} a_j \cdot \cos\frac{(2k-1)j\pi}{2m}$	=	$a_m(z_k)$	where	$z_k = e^{\frac{2k-1}{2m}\pi i}$
DST-I	$\lambda_k = a_0 + 2\sum_{j=1}^{m-1} a_j \cdot \cos\frac{kj\pi}{m+1}$	=	$a_m(z_k)$	where	$z_k = e^{\frac{k}{m+1}\pi i}$
DST-II	$\lambda_k = a_0 + 2\sum_{j=1}^{m-1} a_j \cdot \cos\frac{kj\pi}{m}$	=	$a_m(z_k)$	where	$z_k = e^{\frac{k}{m}\pi i}$
DST-III	$\lambda_k = a_0 + 2\sum_{j=1}^{m-1} a_j \cdot \cos\frac{(2k-1)j\pi}{2m}$	=	$a_m(z_k)$	where	$z_k = e^{\frac{2k-1}{2m}\pi i}$
DST-IV	$\lambda_k = a_0 + 2\sum_{j=1}^{m-1} a_j \cdot \cos\frac{(2k-1)j\pi}{2m}$	=	$a_m(z_k)$	where	$z_k = e^{\frac{2k-1}{2m}\pi i}$

expressions for all eight cases, because we shall use them in our arguments below.) Thus, the eigenvalues $\{\lambda_k\}$ of $S_Q(A)$ are the values of a truncated function

$$a_m(z) = \sum_{k=-(m+1)}^{m+1} a_k z^k$$

at certain points on the unit circle, specified in Table 6. Since $a(x)$ is in the Wiener class, $a_m(x)$ is its approximation, so Property 4 holds true, and further, it implies Properties 5 and 6.

6.3. Terminology: Strang-type preconditioners. Even though the formula for the classical Strang preconditioner

(6.6)
$$S(A_m) = a_0 I + a_1 Z_1 + a_2 Z_1^2 + \cdots + a_2 Z_1^{n-2} + a_1 Z_1^{n-1}, \quad \text{where} \quad Z_1 = \text{ circ } (0, \ldots, 0, 1)$$

looks like (6.1) with H_Q replaced by Z_1 and with $O_j(x)$ replaced by x^{j-1}, it is not immediately clear how strong the analogy is. Moreover, (6.6) has a wrap-around property of $\{a_k\}$ that is missing in the definition given in Table 5.

However, the wrap-around is not a crucial property of the original Strang preconditioner. It is needed only to cope with the nonsymmetry of Z_1 and to make the matrix $S(A_m)$ symmetric. The matrices H_Q are already symmetric, and the wrap-around is not needed in Table 5.

However, $S_Q(A)$ and $S(A)$ share a number of crucial properties, e.g., the trick with computing the spectrum via results as in the Corollary 6.1, deriving Properties 4, 5, and 6, and establishing the equivalence to the T.Chan-type preconditioners. We think these analogies justify the name Strang-type for the new preconditioners.

Summarizing, in this section we presented explicit formulas for Strang-type preconditioners, and proved for them Properties 4–6. Moreover, Properties 1–2 stated in the introduction are also trivially satisfied. It remains only to establish Property 3, crucial for the rapid convergence of the PCGM. This property will be proved in section 8 below, using another description of the new preconditioners given next.

7. The new preconditioners are Toeplitz-plus-Hankel-like matrices.

7.1. The classical Strang (Toeplitz-plus-Toeplitz) preconditioner. We called new preconditioners Strang-like preconditioners; a justification for this nomenclature is offered next. In [S86] Strang proposed a circulant preconditioner,

$$S(A) = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_2 & a_1 \\ a_1 & a_0 & a_1 & \ddots & & a_2 \\ a_2 & a_1 & a_0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_1 & a_2 \\ a_2 & & \ddots & a_1 & a_0 & a_1 \\ a_1 & a_2 & \cdots & a_2 & a_1 & a_0 \end{bmatrix},$$

obtained by copying first $[m/2]$ diagonals of $A = \left[\ a_{|i-j|}\ \right]$. In fact, this preconditioner can be seen as a Toeplitz-plus-Toeplitz matrix,

$$S(A) = A + T,$$

where A is the given Toeplitz matrix, and the first column of a second Toeplitz term T is given by

$$\left[\ 0 \quad \cdots \quad 0 \quad \cdots \quad a_2 - a_{m-2} \quad a_1 - a_{m-1}\ \right]^T.$$

In fact, many favorable properties of $S(A)$ can be explained by the fact that the entries of the central diagonals of A now occupy corner positions in T. In the case when the generating function $a(z) = \sum_{k=-\infty}^{\infty} a_k \cdot z^k$ is from the Wiener class, only the first few coefficients are large, implying that $T = A_{lr} + A_{sn}$ is a sum of a low-rank matrix and a small-norm matrix, a property implying the usefulness of

$$S(A) = A + A_{lr} + A_{sn} \tag{7.1}$$

as a preconditioner for A. It turns out that all eight Strang-type preconditioners $S_Q(A)$ considered above are Toeplitz-plus-Hankel-like matrices (formally defined below), a fact allowing us to use the above low-rank-small-norm-perturbation argument to prove the favorable properties of $S_Q(A)$ as preconditioners for A.

7.2. Toeplitz-plus-Hankel-like matrices. Recall that matrices R with a low ∇_{H_Q}-displacement rank have been called *polynomial Hankel* matrices. Since all eight matrices H_Q in Table 4 correspond to the Chebyshev-like polynomial systems listed in Table 2, we could refer to such R as *Chebyshev–Hankel* matrices. It can be checked, however, that if H_Q is defined as, for example, in the line DST-I of Table 4, then for any sum $T + H$ of a Toeplitz $T = \left[\ t_{i-j}\ \right]$ matrix and a Hankel $H = \left[\ h_{i+j-2}\ \right]$ matrix we have

$$\nabla_{H_Q}(T+H) = \frac{1}{2} \cdot \left(\left[\begin{array}{c|ccc|c} 0 & -t_2 & \cdots & -t_{m-1} & 0 \\ \hline t_2 & & & & t_{m-1} \\ \vdots & & 0 & & \vdots \\ t_{m-1} & & & & t_2 \\ \hline 0 & -t_{m-1} & \cdots & -t_2 & 0 \end{array}\right] + \left[\begin{array}{c|ccc|c} 0 & -h_0 & \cdots & -h_{m-3} & h_{2m-2}-h_m \\ \hline h_0 & & & & h_{m+1} \\ \vdots & & 0 & & \vdots \\ h_{m-3} & & & & h_{2m-2} \\ \hline h_{m-2}-h_m & -h_{m+1} & \cdots & -h_{2m-2} & 0 \end{array}\right] \right).$$

In our terminology, the ∇_{H_Q}-displacement rank of $T+H$ does not exceed four. This fact was observed and used in [HJR88] and [GK89] to develop fast algorithms for inversion of Toeplitz-plus-Hankel matrices. In [GKO95] we introduced (and suggested, for the first time, fast algorithms for) the more general class of Toeplitz-plus-Hankel-like matrices, defined as having low (not just $\alpha \le 4$) ∇_{H_Q}-displacement rank. Clearly (cf. [GKO95]), the other choices for H_Q in Table 4 can be used to define the *same* class of Toeplitz-plus-Hankel-like matrices (the actual displacement rank may vary, depending upon a particular H_Q, but it remains low). Summarizing, there are two nomenclatures (i.e., Chebyshev–Hankel-like and Toeplitz-plus-Hankel-like matrices) for the same class of structured matrices.

7.3. Toeplitz-plus-Hankel-like representations for $S_Q(A)$. Since all the preconditioners $S_Q(A)$ in Table 5 belong to the kernel of the corresponding $\nabla_{H_Q}(\cdot)$, they clearly belong to the above class of Toeplitz-plus-Hankel-like matrices. In fact, each of them can even be represented as

$$S_Q(A) = A + H + B, \tag{7.3}$$

where A is the given Toeplitz matrix, H is a certain Hankel matrix, and B is a certain "border" matrix, having nonzero entries only in its first and last rows and columns.[1]

The proof of the fact that $S_Q(A)$ has the form (7.3), where A is the given Toeplitz matrix, and H, B are specified in Table 7, is based on the following observations.

- The fact that $A+H+B \in \mathcal{K}_Q = \text{ Ker } \nabla_{H_Q}$ can be easily checked by inspection.
- Observe that

$$Q_k(H_Q) \cdot e_1 = Q_0 \cdot e_{k+1}, \tag{7.4}$$

  (cf. [DFZ95]). As was mentioned in sections 4 and 6, matrices $\{Q_k(H_Q)\}$ form a basis in Ker ∇_{H_Q}, and since the first column of the matrix $A+H+B$ coincides with the first column of $S_Q(A)$ given in Table 5, the representations (7.3) follow.

Recall that we started this section saying that all eight $S_Q(A)$ are the (discrete-trigonometric-transform) analogues of the Strang preconditioner S. Indeed, the results in Table 7 show that the form (7.3) for $S_Q(A)$ is similar to that in (7.1) for $S(A)$. Indeed, both the Strang circulant preconditioner $S(A)$ and all of $S_Q(A)$ are constructed by adding to the given Toeplitz matrix A a matrix, in which the entries of the central diagonals of A now occupy the corner locations. This fact is used next to prove the useful properties of $S_Q(A)$.

8. Clustering of the spectrum of $S_Q(A)^{-1}A$. Here we establish the crucial Property 3 for all eight Strang-like preconditioners $S_Q(A)$ under the standard Wiener class assumption. For the preconditioner $S_Q(A)$ corresponding to the DST-I, this property was established in [BB90] and below we adapt their arguments for the other seven $S_Q(A)$. Since $S_Q(A)^{-1}A = I - S_Q(A)^{-1}(H+B)$ with H, B specified in Table 6, it is sufficient to show that the spectrum of $S_Q(A)^{-1}(H+B)$ is clustered around 0. Letting $\varepsilon > 0$ be fixed, choose N such that $\sum_{k=N+1}^{\infty} |a_k| < \varepsilon$. Then we can split

$$H + B = A_{lr} + A_{sn}, \tag{8.1}$$

[1] A reader should be warned that such a specific representation is not valid for arbitrary Toeplitz-plus-Hankel-like matrices.

TABLE 7
A Hankel part and a "border" part of $S_Q(A)$.

	H	B
DCT-I	$\begin{bmatrix} a_0 & a_1 & \cdots & a_{m-2} & 2a_{m-1} \\ a_1 & a_2 & \iddots & \iddots & a_{m-2} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ a_{m-2} & \iddots & \iddots & a_2 & a_1 \\ 2a_{m-1} & a_{m-2} & \cdots & a_1 & a_0 \end{bmatrix}$	$(\sqrt{2}-2)\left[\begin{array}{c\|ccc\|c} -\frac{a_0}{\sqrt{2}-2} & a_1 & \cdots & a_{m-2} & -\frac{a_{m-1}}{\sqrt{2}-2} \\ \hline a_1 & & & & a_{m-1} \\ \vdots & & 0 & & \vdots \\ a_{m-2} & & & & a_1 \\ \hline -\frac{a_{m-1}}{\sqrt{2}-2} & a_{m-2} & \cdots & a_1 & -\frac{a_0}{\sqrt{2}-2} \end{array}\right]$
DCT-II	$\begin{bmatrix} a_1 & a_2 & \cdots & a_{m-1} & 0 \\ a_2 & a_3 & \iddots & \iddots & a_{m-1} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ a_{m-1} & \iddots & \iddots & a_3 & a_2 \\ 0 & a_{m-1} & \cdots & a_2 & a_1 \end{bmatrix}$	0
DCT-III	$\begin{bmatrix} a_0 & a_1 & \cdots & a_{m-2} & a_{m-1} \\ a_1 & a_2 & \iddots & \iddots & 0 \\ \vdots & \iddots & \iddots & \iddots & -a_{m-1} \\ a_{m-2} & \iddots & \iddots & \iddots & \vdots \\ a_{m-1} & 0 & -a_{m-1} & \cdots & -a_2 \end{bmatrix}$	$(\sqrt{2}-2)\left[\begin{array}{c\|ccc} -\frac{a_0}{\sqrt{2}-2} & a_1 & \cdots & a_{m-2} \quad a_{m-1} \\ \hline a_1 & & & \\ \vdots & & 0 & \\ a_{m-2} & & & \\ a_{m-1} & & & \end{array}\right]$
DCT-IV	$\begin{bmatrix} a_1 & a_2 & \cdots & a_{m-1} & 0 \\ a_2 & a_3 & \iddots & \iddots & -a_{m-1} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ a_{m-1} & \iddots & \iddots & -a_3 & -a_2 \\ 0 & -a_{m-1} & \cdots & -a_2 & -a_1 \end{bmatrix}$	0
DST-I	$\begin{bmatrix} -a_2 & \cdots & -a_{m-1} & 0 & 0 \\ \vdots & \iddots & \iddots & \iddots & 0 \\ -a_{m-1} & \iddots & \iddots & \iddots & -a_{m-1} \\ 0 & \iddots & \iddots & \iddots & \vdots \\ 0 & 0 & -a_{m-1} & \cdots & -a_2 \end{bmatrix}$	0
DST-II	$\begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{m-1} & 0 \\ -a_2 & -a_3 & \iddots & \iddots & -a_{m-1} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ -a_{m-1} & \iddots & \iddots & -a_3 & -a_2 \\ 0 & -a_{m-1} & \cdots & -a_2 & -a_1 \end{bmatrix}$	0
DST-III	$\begin{bmatrix} -a_2 & \cdots & -a_{m-1} & 0 & a_{m-1} \\ \vdots & & & \iddots & a_{m-2} \\ -a_{m-1} & \iddots & \iddots & \iddots & \vdots \\ 0 & \iddots & \iddots & a_2 & a_1 \\ a_{m-1} & a_{m-2} & \cdots & a_1 & a_0 \end{bmatrix}$	$\frac{-1}{1+\sqrt{2}}\left[\begin{array}{ccc\|c} & & & a_{m-1} \\ & 0 & & a_{m-2} \\ & & & \vdots \\ & & & a_1 \\ \hline a_{m-1} & a_{m-2} & \cdots & a_1 \quad (1+\frac{1}{\sqrt{2}})a_0 \end{array}\right]$
DST-IV	$\begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{m-1} & 0 \\ -a_2 & -a_3 & \iddots & \iddots & a_{m-1} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ -a_{m-1} & \iddots & \iddots & a_3 & a_2 \\ 0 & a_{m-1} & \cdots & a_2 & a_1 \end{bmatrix}$	0

by taking out in A_{lr} antidiagonals of $H+B$ with the entries $a_0, a_1, \ldots, a_N$. Then the 2-norm of the second matrix in (8.1) can be bounded by 2ε (cf. [BB90]). Hence by the Cauchy interlace theorem, the eigenvalues of $(H+B)$ are clustered around zero, except at most $s =$ rank A_{lr} outliers. Applying the Courant–Fischer theorem to the matrix $S_Q(A)^{-1}(H+B)$, we obtain

$$\lambda_k\{(S_Q(A)^{-1}(H+B)\} < \frac{\lambda_k\{H+B\}}{\min_{|z|=1} f(z)},$$

implying that Property 3 holds.

9. T.Chan-type preconditioners. Here we specify another family of preconditioners $C_Q(A)$, defined by

$$\|C_Q(A) - A\|_F = \min_{R \in \mathcal{K}_Q} \|R - A\|_F,$$

i.e., the optimal Frobenius-norm approximants of A in $\mathcal{K}_Q =$ Ker ∇_{H_Q} (for the circulant case, $H_Q = Z_1$, such a preconditioner was proposed by T. Chan [C88]).

Recall that we designed all Strang-type preconditioners $S_Q(A)$ using a representation of the form (6.1). It turns out that the same basis is convenient for writing down the formulas also for the T.Chan-type preconditioners $C_Q(A)$, and especially for the analysis of the clustering property in section 9. In order to obtain the coefficients in

$$C_Q(A) = \sum_{k=1}^{m} r_k Q_{k-1}(H_Q), \tag{9.1}$$

we solve a linear system of equations

$$\frac{\partial}{\partial r_k}\|C_Q(A) - A\|_F = 0 \qquad (k = 1, 2, \ldots, m). \tag{9.2}$$

To solve (9.2) we found the entries of matrices $\{Q_k(H_Q)\}$ in all eight cases as follows. The entries of the first column of each $Q_k(H_Q)$ are given by (7.4). The entries of the other columns can be recursively computed using the fact that $Q_k(H_Q) \in \mathcal{K}_Q =$ Ker ∇_{H_Q}.

For example, for $n = 5$ we have: $Q_0(H_Q) = Q_0 I$ (with $Q_0 = \frac{1}{\sqrt{2}}$ for DCT-I and DCT-III, and $Q_0 = 1$ in the other six cases), $Q_1(H_Q) = H_Q$ (shown in Table 4), and $Q_2(H_Q), Q_3(H_Q), \ldots$ are given in Table 8.

The idea of using similar bases was also used for DST-I and discrete Hartley transforms in [BB90], [BF93], [BK95].

Using a particular form of $Q_k(H_Q)$ shown in Table 8, we obtain from (9.2) that the desired coefficients in (9.1) are then obtained from the given Toeplitz matrix $A = \left[\ a_{|i-j|}\ \right]$ by

$$\begin{bmatrix} r_1 \\ \vdots \\ r_m \end{bmatrix} = G_Q \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \end{bmatrix}, \tag{9.3}$$

TABLE 8
First-column bases.

	$Q_2(H_Q)$	$Q_3(H_Q)$	$Q_4(H_Q)$
DCT-I	$\begin{bmatrix} 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \end{bmatrix}$
DCT-II	$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 1 & 0 \\ 1 & -1 & 1 & -1 & 1 \\ 0 & 1 & -1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 1 & -1 & 1 & 0 \\ 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$
DCT-III	$\begin{bmatrix} 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & -\frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -\frac{1}{2} \\ 0 & \frac{1}{2} & 0 & -\frac{1}{2} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & \frac{1}{2} & 0 & -\frac{1}{2} & 0 \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{2} & 0 & 0 \end{bmatrix}$
DCT-IV	$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 1 & 0 \\ 1 & -1 & 1 & -1 & 1 \\ 0 & 1 & -1 & 1 & -2 \\ 0 & 0 & 1 & -2 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 1 & -1 & 1 & -2 \\ 1 & -1 & 1 & -2 & 2 \\ 0 & 1 & -2 & 2 & -2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -2 \\ 0 & 0 & 1 & -2 & 2 \\ 0 & 1 & -2 & 2 & -2 \\ 1 & -2 & 2 & -2 & 2 \end{bmatrix}$
DST-I	$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$
DST-II	$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$
DST-III	$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & \sqrt{2} \\ 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & \sqrt{2} & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & \sqrt{2} \\ 0 & 1 & 0 & 2 & 0 \\ 1 & 0 & 2 & 0 & \sqrt{2} \\ 0 & \sqrt{2} & 0 & \sqrt{2} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \sqrt{2} & 0 \\ 0 & 0 & \sqrt{2} & 0 & 1 \\ 0 & \sqrt{2} & 0 & \sqrt{2} & 0 \\ 1 & 0 & 1 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}$
DST-IV	$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 1 & 2 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 2 \\ 1 & 1 & 1 & 2 & 2 \\ 0 & 1 & 2 & 2 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 2 & 2 \\ 0 & 1 & 2 & 2 & 2 \\ 1 & 2 & 2 & 2 & 2 \end{bmatrix}$

where the matrix G_Q has the simple structure shown in Table 9.

TABLE 9
Definition of the T.Chan-type preconditioner $C_Q(A_m)$. Matrix G_Q for (9.3).

DCT-I	$G = \frac{1}{(m-1)^2} \cdot \operatorname{diag}\{\sqrt{2}, 2, 2, \ldots 2, \sqrt{2}\}(D+E+L+U)$, with the terms specified by (9.4), (9.5), (9.6), (9.7)
DCT-II	$\frac{1}{m^2}\begin{bmatrix} m^2 & (m-1)(m-2) & -2(m-2) & \cdots & -4 & -2 \\ 0 & m^2-(m-2) & (m-2)(m-2) & \ddots & -4 & -2 \\ 0 & 2 & m^2-2(m-2) & \ddots & -4 & -2 \\ 0 & 2 & 4 & \ddots & 2(m-2) & -2 \\ 0 & 2 & 4 & \ddots & m^2-(m-2)(m-2) & (m-2) \\ 0 & 2 & 4 & \cdots & 2(m-2) & m^2-(m-1)(m-2) \end{bmatrix}$
DCT-III	$\frac{1}{m}\begin{bmatrix} \sqrt{2}m & & & & & \\ & 2(m+\sqrt{2}-2) & & \mathbf{0} & & \\ & & 2(m+\sqrt{2}-3) & & & \\ & & & \ddots & & \\ & \mathbf{0} & & & 2(\sqrt{2}+1) & \\ & & & & & 2\sqrt{2} \end{bmatrix}$
DCT-IV	$\frac{1}{m}\begin{bmatrix} m & m-1 & & & \\ m-1 & m-2 & & \mathbf{0} & \\ & & m-2 & \ddots & \\ & & \ddots & 2 & \\ & \mathbf{0} & & 2 & 1 \\ & & & & 1 \end{bmatrix}$

(9.4)

$$D = \operatorname{diag}\Big((m-1)^2, \boxed{2\sqrt{2}(m-1)+(m-3)(m-3)}, \boxed{2\sqrt{2}(m-1)+(m-3)(m-4)}, \ldots$$

$$\ldots, \boxed{2\sqrt{2}(m-1)+2(m-3)}, \boxed{2\sqrt{2}(m-1)+(m-3)}, \boxed{2\sqrt{2}(m-1)}, (2m-3)\Big)$$

(a recursion for the $2, 3, \ldots, m-2, m-1$ entries is apparent).

(9.5)

$$E = -2\sqrt{2}\ \text{toeplitz}\left(\left[\ 1,\quad 0,\quad 1,\quad 0,\quad 1,\quad 0,\quad \ldots\ \right]\right) \cdot \operatorname{diag}\left(\left[\ 0,\quad 1,\quad 1,\quad \ldots, 1,\quad 0\ \right]\right).$$

Here we follow the MATLAB notations, where toeplitz (c, r) denotes the Toeplitz matrix with the first column c and the first row r. toeplitz (c) denotes the symmetric Toeplitz matrix with the first column c.

(9.6)

$$L = \text{toeplitz}\left(\left[\ 0,\quad 0,\quad 1,\quad 0,\quad 1,\quad 0,\quad 1,\quad 0,\quad \ldots\ \right], \left[\ 0,\quad 0,\quad 0,\quad \ldots\ \right]\right) \times \operatorname{diag}\left(\left[\ 0,\quad 2\cdot 2,\quad 2\cdot 3,\quad \ldots,\quad 2\cdot(m-2),\quad 0,\quad 0\ \right]\right),$$

(9.7)

$$U = \text{toeplitz}\left(\left[\ 0,\quad 0,\quad 0,\quad \ldots\ \right], \left[\ 0,\quad 0,\quad 1,\quad 0,\quad 1,\quad 0,\quad 1,\quad 0,\quad \ldots\ \right]\right) \times \operatorname{diag}\left(\left[\ 0,\quad 0,\quad -2(m-4),\quad -2(m-3),\quad \ldots,\quad -4,\quad -2,\quad 0,\quad -1\ \right]\right).$$

TABLE 9 (*cont.*)
Definition of the T.Chan-type preconditioner $C_Q(A_m)$. Matrix G_Q for (9.3).

DST-I	$\frac{1}{m+1}\begin{bmatrix} m+1 & 0 & -(m-2) & & & & \\ & m+1 & 0 & -(m-3) & & & \\ & & m & \ddots & \ddots & & \\ & & & \ddots & 0 & -2 & \\ & \mathbf{0} & & & 5 & 0 & -1 \\ & & & & & 4 & 0 \\ & & & & & & 3 \end{bmatrix}$
DST-II	$\frac{1}{m^2}\begin{bmatrix} m^2 & -(m-1)(m-2) & -2(m-2) & \cdots & (-1)^{m-2}4 & (-1)^{m-1} \\ 0 & m^2-(m-2) & -(m-2)(m-2) & \ddots & \vdots & \vdots \\ 0 & -2 & m^2-2(m-2) & \ddots & -4 & 2 \\ 0 & 2 & -4 & \ddots & -2(m-2) & -2 \\ 0 & \vdots & \vdots & \ddots & m^2-(m-2)(m-2) & -(m-2) \\ 0 & (-1)^m 2 & (-1)^{m-1}4 & \cdots & -2(m-2) & m^2-(m-1)(m-2) \end{bmatrix}$
DST-III	$\frac{1}{m}\begin{bmatrix} m & 0 & -(-3+\sqrt{2}) & & & & \\ & m-2+\sqrt{2} & 0 & \ddots & & & \\ & & m-3+\sqrt{2} & \ddots & -(2+\sqrt{2}) & & \\ & & & \ddots & 0 & -(1+\sqrt{2}) & \\ & \mathbf{0} & & & 2+\sqrt{2} & 0 & -\sqrt{2} \\ & & & & & 1+\sqrt{2} & 0 \\ & & & & & & 2 \end{bmatrix}$
DST-IV	$\frac{1}{m}\begin{bmatrix} m & -(m-1) & & & \\ & m-1 & -(m-2) & & \mathbf{0} \\ & & m-2 & \ddots & \\ & & & \ddots & -2 \\ & \mathbf{0} & & & 2 \quad -1 \\ & & & & 1 \end{bmatrix}$

In [CNW96] a method is proposed to find the optimal sine transform based preconditioner when we have the corresponding basis.

Formula (9.3) along with the data in Table 9 shows that all T.Chan-type preconditioners trivially satisfy Properties 1 and 2. In the next section we establish Property 3 for these preconditioners.

10. Clustering of eigenvalues of $C_Q(A)^{-1}A$.

10.1. Clustering. Here we establish for all eight preconditioners $C_Q(A)$ Property 3 under the Wiener class assumption, by showing that the spectra of $S_Q(A)$ and $C_Q(A)$ are asymptotically the same (for the classical Strang and T.Chan circulant preconditioners, this property was established in [C89]).

PROPOSITION 10.1. *Let A_m be a finite section of a semi-infinite Toeplitz matrix, let $\{Q_k\}_{k=0}^{m-1}$ be one of the eight polynomial systems in Table* 2, *and let $S_Q(A), C_Q(A) \in \mathbb{R}^{n\times n}$ denote the corresponding Strang-type and the T.Chan-type preconditioners, respectively. Then*

$$\lim_{m\to\infty} \|C_Q(A) - S_Q(A)\|_2 = 0,$$

where $\|\cdot\|_2$ denotes the spectral norm in $\mathbb{R}^{m\times m}$.

Proof. Since both $C_Q(A)$ and $S_Q(A)$ belong to $\mathcal{K}_Q$, they are both diagonalized by the corresponding discrete-trigonometric-transform matrix T_Q; see, e.g., Proposition 6.1. Since T_Q is an orthogonal matrix (i.e, one of the eight orthogonal matrices displayed in Table 1), essentially we have to establish the convergence to zero of the eigenvalues of $S_Q(A) - C_Q(A)$:

$$\lim_{m\to\infty} \lambda_k(S_Q(A) - C_Q(A)) = 0 \qquad (k = 1, 2, \ldots, m).$$

Again, by Proposition 6.1 the eigenvalues of $S_Q(A)$ and $C_Q(A)$ can be obtained from the coefficients $\{r_k\}$ in the representation (6.2) for these matrices by using (6.5). As shown in section 9, for the $C_Q(A)$ these coefficients are given by

$$\begin{bmatrix} r_1 \\ \vdots \\ r_m \end{bmatrix} = G_Q \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \end{bmatrix}, \tag{10.1}$$

where the matrices G_Q are listed in Table 9. For convenience we next rewrite the results of Table 5 in a similar manner. Moreover, the coefficients in the representation $S_Q(A) = \sum_{k=0}^{m-1} r_k C_Q(H_Q^T)$ are obtained by

$$\begin{bmatrix} r_1 \\ \vdots \\ r_m \end{bmatrix} = R_Q \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \end{bmatrix}, \tag{10.2}$$

where matrices R_Q are specified in Table 10.

By comparing (6.5), (10.1), and (10.2) we have

$$\begin{bmatrix} \lambda_1(S_Q(A) - C_Q(A)) \\ \vdots \\ \lambda_m(S_Q(A) - C_Q(A)) \end{bmatrix} = V_Q \cdot (R_Q - G_Q) \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \end{bmatrix}, \tag{10.3}$$

where the matrices G_Q and R_Q are displayed in Tables 9 and 10, respectively. Recall that not all of the eight matrices V_Q have uniformly bounded entries; see, e.g., Table 2. Therefore it is more convenient to rewrite (10.3) as

$$\begin{bmatrix} \lambda_1(S_Q(A) - C_Q(A)) \\ \vdots \\ \lambda_m(S_Q(A) - C_Q(A)) \end{bmatrix} = (V_Q R_Q) \cdot (I - R_Q^{-1} G_Q) D^{-1} \cdot \left(D \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \end{bmatrix} \right), \tag{10.4}$$

where $D = \text{ diag } (\frac{1}{m}, \frac{2}{m}, \ldots, \frac{m-1}{m}, 1)$. Now we can prove the statement of the proposition, i.e., that the entries on the left-hand side of (10.4) tend to zero, by making the following three observations for three factors on the right-hand side of (10.4).

1. *Left factor.* The entries of the matrix $V_Q R_Q$ are uniformly bounded independently of m.
2. *Middle factor.* The column sums of the matrix $(I - R_Q^{-1} G_Q) D^{-1}$ have uniformly bounded column sums.
3. *Right factor.* If $f(z) = \sum_{k=-\infty}^{\infty} a_k z^k$ is from the Wiener class, then $\forall \varepsilon > 0$ $\exists N > 1$ such that $\forall M > N$ we have

$$\sum_{k=0}^{M} \frac{k}{M} |a_k| < \varepsilon.$$

TABLE 10
Definition of $S_Q(A_m)$. The matrix R_Q in (10.2).

DCT-I	$\begin{bmatrix} \sqrt{2} & & & & & \\ & 2 & & & & \\ & & 2 & & & \\ & & & \ddots & & \\ & & & & 2 & \\ & & & & & 2\sqrt{2} \end{bmatrix}$	DST-I	$\begin{bmatrix} 1 & 0 & -1 & & & & \\ & 1 & 0 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & -1 & \\ & & & & 1 & 0 & -1 \\ & & & & & 1 & 0 \\ & & & & & & 1 \end{bmatrix}$
DCT-II	$\begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & 1 \\ & & & & & 1 \end{bmatrix}$	DST-II	$\begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & -1 \\ & & & & & 1 \end{bmatrix}$
DCT-III	$\begin{bmatrix} \sqrt{2} & & & & & \\ & 2 & & & & \\ & & 2 & & & \\ & & & \ddots & & \\ & & & & 2 & \\ & & & & & 2 \end{bmatrix}$	DST-III	$\begin{bmatrix} 1 & 0 & -1 & & & & \\ & 1 & 0 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & -1 & \\ & & & & 1 & 0 & -1 \\ & & & & & 1 & 0 \\ & & & & & & \sqrt{2} \end{bmatrix}$
DCT-IV	$\begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & 1 \\ & & & & & 1 \end{bmatrix}$	DST-IV	$\begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & -1 \\ & & & & & 1 \end{bmatrix}$

The first observation can be deduced from the comparison of (10.2) and Table 6, showing that $V_Q \cdot R_Q$ is a "cosine" matrix.

The assertion in the second observation is deduced from the particular form of matrices R_Q and G_Q displayed in Tables 9 and 10. The arguments are immediate in the cases DCT-III, DCT-IV, DST-III, and DST-IV, and they are not much more involved in the case of DST-I. In the cases of DCT-I, DCT-II, and DST-II one has to split G_Q into three parts: bidiagonal, and upper and lower triangular, and for each of the corresponding parts of $(I - R_Q^{-1}G_Q)D^{-1}$ the statement is easily deduced. The third observation is immediate (cf. [C89]).

Thus, Proposition 10.1 is now proved, and it implies Property 3 using standard arguments (cf. [C89]). □

10.2. Transformation-to-Cauchy-like and the Tyrtyshnikov property. In the previous section we proved Properties 1–3 for $C_Q(A_m)$. Properties 4 and 6 (formulated in section 6) also follow from Proposition 10.1. Here we show that Property 5 also holds, and, moreover, we prove that for any of eight T.Chan-type preconditioners the following *Tyrtyshnikov property* holds independently of m:

$$\lambda_{min}(A_m) \le \lambda_{min}(C_Q(A_m)) \le \lambda_{max}(C_Q(A_m)) \le \lambda_{max}(A_m). \tag{10.5}$$

TABLE 11
Transformation-to-Cauchy-like.

	Toeplitz	→	Cauchy-like
Matrix	A_m	→	$T_Q \cdot A_m \cdot T_Q^T$
Generator	G_Q	→	$T_Q \cdot G_Q$
	J	→	J
	$C_Q(A_m)$	→	$T_Q \cdot C_Q(A_m) \cdot T_Q^T$

For the circulant T.Chan preconditioner such a property was proved in [T92] and [CJY91].

To prove (10.5) we shall use our definitions in section 2. Observe that since the Frobenius norm in $\mathbb{R}^{m\times m}$ generates the inner product (2.6), we have that the ∇_{H_Q}-generator of A_m is given by $\{G_Q, J, C_Q(A_m)\}$ in

$$H_Q A_m - A_m H_Q = G_Q J G_Q^T.$$

The particular form of the $m \times 4$ matrix G_Q and 4×4 matrix J is not relevant at the moment (for each H_Q of Table 4 they can be easily written down as in, for example, (7.2). It is important that the corresponding T.Chan-type preconditioner describes the kernel component of A_m, i.e., the third matrix in its ∇_{H_Q}-generator. Furthermore, specifying the "transformation-to-Cauchy-like" Proposition 4.1 to our settings here we obtain the results displayed in Table 11.

The inequality (10.5) now follows from the following two observations. First, since T_Q is orthogonal, the spectra of A_m and $T_Q A_m T_Q^T$ and of $C_Q(A_m)$ and $T_Q C_Q(A_m) T_Q^T$ are, respectively, the same. Second, since the Frobenius norm is unitary-equivalent, we have that the diagonal matrix $T_Q \cdot C_Q(A_m) \cdot T_Q^T$ is the optimal Frobenius-norm diagonal approximants of the Cauchy-like matrix $T_Q A_m T_Q^T$. In other words, $T_Q \cdot C_Q(A_m) \cdot T_Q^T$ is simply a diagonal part of $T_Q A_m T_Q^T$, implying (10.5).

The above arguments indicate that there is a close connection between finding an optimal Frobenius-norm approximant of a Toeplitz matrix and transformations of Toeplitz matrices to Cauchy-like matrices. Such transformations were closely studied in several recent papers. For example, in [O93b], [O95], and [He95b] direct (i.e., based on the computation on the matrix entries, without explicit use of displacement operators) transformations were considered. Their results can be applied to obtain T.Chan-type preconditioners for the transforms II and III.

In this paper explicit formulas are obtained not only for T.Chan-type, but also for Strang-type preconditioners. Moreover, in obtaining T.Chan-type preconditioners we follow [GO94a], [GKO95], [KO95a], [KO94], [KO95b], and explore a different approach to transformations to Cauchy-like matrices. The crucial point here is to introduce an appropriate displacement operator, ∇_{H_Q}, where H_Q is diagonalized by a unitary matrix. New transformation formulas (systematically obtained here for all eight cases) require only one discrete trigonometric transform to compute the diagonal part of a Cauchy-like matrix, as compared to two such transforms in [O93b], [O95], [He95b]. Furthermore, the concept of partially reconstructible matrices suggested using the definition of ∇_{H_Q}-generator, given in [KO95a]. This allowed us to obtain unified descriptions for both the Strang-type and T.Chan-type preconditioners, given in Tables 9 and 10, respectively. These new formulas allowed us to establish in sec-

tion 9 the result on close asymptotic behavior of both classes of preconditioners, and to prove the crucial clustering property for $C_Q(A_m)$.

11. Real-arithmetic algorithms for multiplication of a Toeplitz matrix by a vector. Embedding.

11.1. Real symmetric Toeplitz matrices. In the first part of the paper we developed two families of Strang-type and T.Chan-type preconditioners for real symmetric Toeplitz matrices, and established Properties 1–3, guaranteeing a convergence for the PCGM (under the Wiener class assumption). In this and the next sections we address the question of how to efficiently organize the iteration process itself.

First observe that all the computations with new preconditioners (i.e., their construction and then solving the associated linear systems) can be done in real arithmetic. To fully exploit this advantageous property we have to specify an efficient real-arithmetic algorithm for multiplication of a Toeplitz matrix by a vector (the standard technique is based on the FFT, assuming complex arithmetic). In this section we observe that the explicit formulas obtained for the Strang-type readily suggest such algorithms for all eight cases. These algorithms can be derived by the following two steps.

- First, we embed an $m \times m$ Toeplitz matrix A_m into a larger $2m \times 2m$ Toeplitz matrix $\mathcal{A}_{2m}$ by padding its first column with m zeros.
- Second, we construct for $\mathcal{A}_{2m}$ the Strang-type preconditioner $S_Q(\mathcal{A}_{2m})$.

As was shown in section 7, this preconditioner admits a Toeplitz-plus-Hankel-plus-border representation,

$$S_Q(\mathcal{A}_{2m}) = \mathcal{A}_{2m} + \mathcal{H} + \mathcal{B},$$

with the Hankel part $\mathcal{H}$ and the "border" part $\mathcal{B}$ displayed in Table 7. Now, taking into account the banded structure of $\mathcal{A}_{2m}$, one sees that in all eight cases the Hankel and the "border" part do not affect the central part of $S_Q(\mathcal{A}_{2m})$. In other words, our initial matrix A_m is a submatrix of $S_Q(\mathcal{A}_{2m})$. This observation allows us to use any of eight DCTs or DSTs to multiply real symmetric Toeplitz matrices by a vector in only two discrete trigonometric transforms of the order $2m$ (one more such transform is needed to compute the diagonal form for $S_Q(A)$). For the case DST-I such an algorithm was proposed earlier by Boman and Koltracht in [BK95].

Although this is beyond our needs in the present paper, note that the formulas for the Strang-type preconditioners allow us to multiply in the same way Toeplitz-plus-Hankel matrices by a vector. These algorithms are analogues of the well-known embedding-into-circulant (complex arithmetic) multiplication algorithm. There is another well-known (complex arithmetic) method for multiplying a Toeplitz matrix A_m by a vector, based on the decomposition of A_m into a sum of circulant and skew-circulant matrices. Real-arithmetic discrete-trigonometric-transform based analogues of this algorithm are offered in section 12. However, before presenting them it is worth noting that the formulas for the Strang-type preconditioners also allow a multiplication of a nonsymmetric Toeplitz by a vector.

11.2. Nonsymmetric Toeplitz matrices. If A is a nonsymmetric Toeplitz matrix, it can be embedded into a symmetric Toeplitz matrix $\left[\begin{smallmatrix} B & A \\ A^* & B \end{smallmatrix}\right]$ and the latter can be used to multiply A by vectors.

TABLE 12
Decompositions for II *and* IV *transforms.*

DCT-II	$T_{C2}A_mT_{C2}^T = D_{C2} + T_{C2}T_{S2}^T D_{S2}T_{S2}T_{C2}^T$
DST-II	$T_{S2}A_mT_{S2}^T = T_{S2}T_{C2}^T D_{C_2}T_{C2}T_{S2}^T + D_{S2}$
DCT-IV	$T_{C4}A_mT_{C4}^T = D_{C4} + T_{C4}T_{S4}^T D_{S4}T_{S4}T_{C4}^T$
DST-IV	$T_{S4}A_mT_{S4}^T = T_{S4}T_{C4}^T D_{C_4}T_{C4}T_{S4}^T + D_{S4}$

12. Another way to compute the matrix-vector product. Decomposition. In this section we describe alternative algorithms to multiply a real symmetric Toeplitz matrix by a vector. These new algorithms are based on the formulas, which are counterparts of the well-know decomposition of a Toeplitz matrix into a sum of a circulant and a skew-circulant matrix. For example, from the Toeplitz-plus-Hankel-plus-border decompositions in Table 7 it immediately follows that

$$A_m = \frac{1}{2}(S_{C2}(A_m) + S_{S2}(A_m)), \qquad A_m = \frac{1}{2}(S_{C4}(A_m) + S_{S4}(A_m)), \tag{12.1}$$

where we denote by $C1, C2, C3, C4, S1, S2, S3, S4$ the corresponding polynomial systems Q of Table 2. Since the preconditioners $S_Q(A_m)$ are diagonalized by the corresponding transform matrices T_Q, each of these formulas clearly allows us to multiply A_m by a vector in just four discrete trigonometric transforms (with two more transforms needed only once to diagonalize $S_Q(A)$).

13. Transformation-to-Cauchy-like approach for the PCGM. As was detailed in section 4, polynomial Hankel-like matrices (this class includes Toeplitz-like matrices) can be transformed into Cauchy-like matrices. In several recent papers this idea has been found to be useful for design of accurate *direct* methods for solving Toeplitz linear systems.

In this section we suggest an application of this technique to PCGM for Toeplitz matrices. More specifically, instead of applying PCGM to the preconditioned system

$$S_Q(A_m)^{-1}A_m x = b, \tag{13.1}$$

we suggest applying it to the *transformed system*

$$(T_Q S_Q(A_m)^{-1}T_Q^T) \cdot (T_Q A_m T_Q^T)(T_Q x) = T_Q b,$$

where the preconditioner is transformed into the diagonal matrix $(T_Q S_Q(A_m)^{-1}T_Q^T)$, and the Toeplitz matrix A_m is transformed into a Cauchy-like matrix $T_Q A_m T_Q^T$. Since a diagonal linear system can be solved in m operations, such a transformation saves us two discrete transforms per iteration, if we can multiply the Cauchy-like matrix $T_Q A_m T_Q^T$ by a vector with exactly the same complexity as the initial matrix A_m.

The formulas in Table 12 allow us to do so, requiring only four real discrete trigonometric transforms per iteration (cf. [H94] for FFT).

Here

$$D_Q = W_Q T_Q R_Q \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \end{bmatrix}, \tag{13.2}$$

TABLE 13
Continuation. Decompositions for the transforms of the type I.

DCT-I	$T_{C1}A_mT_{C1}^T = D_{C1} + T_{C1}(T_{S1}^T D_{S1} T_{S1} + B_{C1})T_{C1}^T$
DST-I	$T_{S1}A_mT_{S1}^T = T_{S1}(T_{C1}^T D_{C1} T_{C1} + B_{C1})T_{S1}^T + D_{S1}$

where the matrices R_Q are displayed in Table 10. These formulas reduce the complexity of one iteration to four real discrete trigonometric transforms of the order m, as compared to the six such transforms of the methods in the previous section.

13.1. Discrete transforms I. Thus for the II and IV transforms the formulas (12.1) seem to be simple because in these cases the Hankel parts of the corresponding cosine and sine Strang-type preconditioners differ only by the sign (see, e.g., Table 7). For the I and III transforms this is not so, and the reason seem to be that the definitions of the corresponding discrete transforms are not chosen to imply for them the representations of the form (12.1). However, instead of changing the standard definitions (for example, taking care of different $N+1$ and $N-1$ and of the size for the DCT-I, DST-I, DCT-III, and DST-III), we show that even with standard definitions in the remaining two cases one can derive not much more involved formulas, also leading to the same efficiency of four discrete transforms per iteration.

Indeed, in the case of DCT-I and DST-I we have the following. Let the numbers $\{c_k\}$, $\{s_k\}$ be defined by

$$(I + (Z^T)^2)\begin{bmatrix} f_0 \\ \vdots \\ f_{n-1} \end{bmatrix} = \begin{bmatrix} a_0 \\ \vdots \\ a_{n-1} \end{bmatrix}, \qquad \begin{bmatrix} e_0 \\ \vdots \\ e_{n-1} \end{bmatrix} = (Z^T)^2 \begin{bmatrix} f_0 \\ \vdots \\ f_{n-1} \end{bmatrix},$$

where Z denotes the lower shift matrix. Then clearly

$$A = S_{C1}(E_m) + S_{S1}(F_m) - B_{C1},$$

where $S_{C1}(E_m)$, $S_{S1}(F_m)$ are Strang-type preconditioners from Table 7 for Toeplitz matrices E_m and F_m defined by their first columns $[e_k]$ and $[f_k]$, respectively. The matrix B_{C1} is the border matrix of F_m defined in the row DCT-I of the same table. Therefore we have the formulas in Table 13.

Here all the diagonal matrices are obtained by (13.2) with the replacement of $[a_k]$ by the $[f_k]$ and $[e_k]$, respectively. Since B_{C1} is the rank-four matrix, these formulas allow us to compute the product of $T_Q A_m$ by a vector in four real trigonometric transforms of the order m. Note that a different formula of this kind was obtained for DST-I in [H95].

13.2. Discrete transforms III. In this case we have

$$A_m = \frac{1}{2}(S_{C3}(A_m) - B_{C3} + ZS_{S3}(A_m)Z^T),$$

$$A_m = \frac{1}{2}(Z^T S_{C3}(A_m)Z + S_{S3}(A_m) - B_{S3}),$$

leading to the formulas in Table 14.

TABLE 14
Continuation. Decompositions for the transforms of type III.

DCT-III	$T_{C3}A_mT_{C3}^T = \frac{1}{2}(D_{C3} + T_{C3}(ZT_{S3}^T D_{S3}T_{S3}Z^T - B_{C3})T_{C3}^T)$
DST-III	$T_{S3}A_mT_{S3}^T = \frac{1}{2}(T_{S1}(Z^T T_{C3}^T D_{C3}T_{C3}Z - B_{S3})T_{S3}^T + D_{S3})$

Acknowledgment. The authors would like to thank the editor, Franklin Luk, for his patience and for making a number of helpful suggestions.

REFERENCES

[BB90] D. BINI AND F. DI BENEDETTO, *A new preconditioner for the parallel solution of positive definite Toeplitz systems*, in Proceedings of the Second ACM Symposium on Parallel Algorithms and Architectures, Crete, Greece, 1990, pp. 220–223.

[BC83] D. BINI AND M. CAPOVANI, *Spectral and computational properties of band symmetric Toeplitz matrices*, Linear Algebra Appl., 52 (1983), pp. 99–126.

[BF93] D. BINI AND P. FAVATI, *On a matrix algebra related to the discrete Hartley transform*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 500–507.

[BK95] E. BOMAN AND I. KOLTRACHT, *Fast transform based preconditioners for Toeplitz equations*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 628–645.

[BKO94] T. BOROS, T. KAILATH, AND V. OLSHEVSKY, *Pivoting and backward stability of fast algorithm for solving Cauchy linear equations*, Linear Algebra Appl., 343/344 (2002), pp. 63–99.

[C88] T. F. CHAN, *An optimal circulant preconditioner for Toeplitz systems*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 766–771.

[C89] R. H. CHAN, *Circulant preconditioners for Hermitian Toeplitz systems*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 542–550.

[CJY91] R. CHAN, X. JIN, AND M. YEUNG, *The circulant operator in the Banach algebra of matrices*, Linear Algebra Appl., 149 (1991), pp. 41–53.

[CNP94] R. CHAN, J. NAGY, AND R. PLEMMONS, *Displacement preconditioner for Toeplitz least squares iteration*, Electron. Trans. Numer. Anal., 2 (1994), pp. 44–56.

[CNW96] R. CHAN, M. NG, AND C. WONG, *Sine transform based preconditioners for symmetric Toeplitz systems,* Linear Algebra Appl., 232 (1996), pp. 237–259.

[CS89] R. H. CHAN AND G. STRANG, *Toeplitz equations by conjugate gradients with circulant preconditioner*, SIAM J. Sci. Stat. Comput., 10 (1989), pp. 104–119.

[CY92] R. CHAN AND M. YENG, *Circulant preconditioners for Toeplitz matrices with positive continuous generating functions*, Math. Comp., 58 (1992), pp. 233–240.

[CN96] R. H. CHAN AND M. K. NG, *Conjugate gradient methods for Toeplitz systems*, SIAM Rev., 38 (1996), pp. 427–482.

[DFZ95] C. DI FIORE AND P. ZELLINI, *Matrix decompositions using displacement rank and classes of commutative matrix algebras*, Linear Algebra Appl., 229 (1995), pp. 49–99.

[GK89] I. GOHBERG AND I. KOLTRACHT, *Efficient algorithm for Toeplitz plus Hankel matrices*, Integral Equations Operator Theory, 12 (1989), pp. 136–142.

[GKO95] 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.

[GL89] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[GO94a] I. GOHBERG AND V. OLSHEVSKY, *Complexity of multiplication with vectors for structured matrices*, Linear Algebra Appl., 202 (1994), pp. 163–192.

[GO94b] 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.

[GS84] U. GRENADER AND G. SZEGÖ, *Toeplitz Forms and Their Applications*, 2nd ed., Chelsea, New York, 1984.

[Gu98] M. GU, *Stable and efficient algorithms for structured systems of linear equations*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 279–306.

[He95a] G. Heinig, *Inversion of generalized Cauchy matrices and other classes of structured matrices*, in Linear Algebra Signal Processing, IMA Vol. Math. Appl. 69, Springer-Verlag, New York, 1995, pp. 63–81.

[He95b] G. Heinig and A. Bojanczyk, *Transformation techniques for Toeplitz and Toeplitz-plus-Hankel matrices. I. Transformations*, Linear Algebra Appl., 254 (1997), pp. 193–226.

[H94] T. Huckle, *Iterative methods for Toeplitz matrices*, Report SCCM-94-05, Computer Science Department, Stanford University, Stanford, CA, 1994.

[H95] T. Huckle, *Cauchy matrices and iterative methods for Toeplitz matrices*, Proc. SPIE, 2563 (1995), pp. 281–292.

[HJR88] G. Heinig, P. Jankowski, and K. Rost, *Fast inversion of Toeplitz-plus-Hankel matrices*, Numer. Math., 52 (1988), pp. 665–682.

[K87] T. Kailath, *Signal processing applications of some moment problems*, in Moments in Mathematics (San Antonio, TX, 1987), Proc. Sympos. Appl. Math. 37, H. Landau, ed., 1987, pp. 71–109.

[KKM79] T. Kailath, S. Kung, and M. Morf, *Displacement ranks of matrices and linear equations*, J. Math. Anal. Appl., 68 (1979), pp. 395–407.

[KO94] T. Kailath and V. Olshevsky, *Displacement structure approach to polynomial Vandermonde and related matrices*, Linear Algebra Appl., 261 (1997), pp. 49–90.

[KO95a] T. Kailath and V. Olshevsky, *Diagonal pivoting for partially reconstructible Cauchy-like matrices, with applications to Toeplitz-like linear equations and to boundary rational matrix interpolation problems*, Linear Algebra Appl., 254 (1997), pp. 251–302.

[KO95b] T. Kailath and V. Olshevsky, *Displacement structure approach to Chebyshev-Vandermonde and related matrices*, Integral Equations Operator Theory, 22 (1995), pp. 65–92.

[KO96] T. Kailath and V. Olshevsky, *Displacement structure approach to discrete-trigonometric-transform based preconditioners of G.Strang type and of T.Chan type*, Calcolo, 33 (1996), pp. 191–208.

[MB79] J. Maroulas and S. Barnett, *Polynomials with respect to a general basis. I. Theory*, J. Math. Anal. Appl., 72 (1979), pp. 177–194.

[O93a] V. Olshevsky, *A fast real-arithmetic algorithm for multiplication of a Toeplitz matrix by a vector*, preprint, Stanford University, Stanford, CA, 1993.

[O93b] M. Ohsmann, *Fast cosine transform of Toeplitz matrices, algorithm and applications*, IEEE Trans. Signal Process., 41 (1993), pp. 3057–3061.

[O95] M. Ohsmann, *Fast transforms of Toeplitz matrices*, Linear Algebra Appl., 231 (1995), pp. 181–192.

[O03] V. Olshevsky, *Pivoting for structured matrices and rational tangential interpolation*, in Fast Algorithms for Structured Matrices: Theory and Applications (South Hadley, MA, 2001), Contemp. Math. 323, AMS, Providence, RI, 2003, pp. 1–75.

[O04] V. Olshevsky, *A displacement structure approach to the derivation of the eight versions of fast cosine and sine transforms*, in preparation.

[OOW03] A. Olshevsky, V. Olshevsky, and J. Wang, *A comrade-matrix-based derivation of the eight versions of fast cosine and sine transforms*, in Fast Algorithms for Structured Matrices: Theory and Applications (South Hadley, MA, 2001), Contemp. Math. 323, AMS, Providence, RI, 2003, pp. 119–149.

[P90] V. Pan, *On computations with dense structured matrices*, Math. Comp., 55 (1990), pp. 179–190.

[S86] G. Strang, *A proposal for Toeplitz matrix calculations*, Stud. Appl. Math., 74 (1986), pp. 171–176.

[SB95] D. Sweet and R. Brent, *Error analysis of a partial pivoting method for structured matrices*, Advanced Signal Processing Algorithms, Proc. SPIE, 2563, 1995, pp. 266–280.

[T92] E. E. Tyrtyshnikov, *Optimal and superoptimal circulant preconditioners*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 459–473.

[T95] E. Tyrtyshnikov, *Circulant preconditioners with unbounded inverses*, Linear Algebra Appl., 216 (1995), pp. 1–24.

[TS96] E. Tyrtyshnikov and V. Strela, *Which preconditioner is better?*, Math. Comp., 65 (1996), pp. 137–150.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 3, pp. 735–747

A DIFFERENTIAL GEOMETRIC APPROACH TO THE GEOMETRIC MEAN OF SYMMETRIC POSITIVE-DEFINITE MATRICES*

MAHER MOAKHER†

Abstract. In this paper we introduce metric-based means for the space of positive-definite matrices. The mean associated with the Euclidean metric of the ambient space is the usual arithmetic mean. The mean associated with the Riemannian metric corresponds to the geometric mean. We discuss some invariance properties of the Riemannian mean and we use differential geometric tools to give a characterization of this mean.

Key words. geometric mean, positive-definite symmetric matrices, Riemannian distance, geodesics

AMS subject classifications. 47A64, 26E60, 15A48, 15A57

DOI. 10.1137/S0895479803436937

1. Introduction.

Almost 2500 years ago, the ancient Greeks defined a list of 10 (actually 11) distinct "means" [14, 23]. All these means are constructed using geometric proportions. Among these are the well-known arithmetic, geometric, and harmonic (originally called "subcontrary") means. These three principal means, which are used particularly in the works of Nicomachus of Gerasa and Pappus, are the only ones out of the original 11 that are still commonly used.

The arithmetic, geometric, and harmonic means, originally defined for two positive numbers, generalize naturally to a finite set of positive numbers. In fact, for a set of m positive numbers, $\{x_k\}_{1\leq k\leq m}$, the arithmetic mean is the positive number $\bar{x} = \frac{1}{m}\sum_{k=1}^{m} x_k$. The arithmetic mean has a variational property; it minimizes the sum of the squared distances to the given points x_k,

$$\bar{x} = \arg\min_{x>0} \sum_{k=1}^{m} \mathrm{d}_e(x, x_k)^2, \tag{1.1}$$

where $\mathrm{d}_e(x, y) = |x - y|$ represents the usual Euclidean distance in $\mathbb{R}$. The geometric mean of $x_1, \ldots, x_m$, which is given by $\tilde{x} = \sqrt[m]{x_1 x_2 \cdots x_k}$, also has a variational property; it minimizes the sum of the squared *hyperbolic distances* to the given points x_k,

$$\tilde{x} = \arg\min_{x>0} \sum_{k=1}^{m} \mathrm{d}_h(x_k, x)^2, \tag{1.2}$$

where $\mathrm{d}_h(x, y) = |\log x - \log y|$ is the hyperbolic distance[1] between x and y. The harmonic mean of the set of m positive numbers $\{x_k\}_{1\leq k\leq m}$ is simply given by the

*Received by the editors October 27, 2003; accepted for publication (in revised form) by U. Helmke June 12, 2004; published electronically April 8, 2005. This work was partially supported by the Swiss National Science Foundation.

http://www.siam.org/journals/simax/26-3/43693.html

†Laboratory for Mathematical and Numerical Modeling in Engineering Science, National Engineering School at Tunis, Tunis El-Manar University, ENIT-LAMSIN, B.P. 37, 1002 Tunis-Belvédère, Tunisia (Maher.Moakher@enit.rnu.tn).

[1]We borrow this terminology from the hyperbolic geometry of the Poincaré upper half-plane. In fact, the hyperbolic length of the geodesic segment joining the points $P(a, y_1)$ and $Q(a, y_2)$, $y_1, y_2 > 0$, is $|\log \frac{y_1}{y_2}|$ (see [24, 27]).

inverse of the arithmetic mean of their inverses, i.e., $\hat{x} = [\frac{1}{m}\sum_{k=1}^{m}(x_k)^{-1}]^{-1}$, and thus the harmonic mean has a variational characterization as well.

The arithmetic mean has been widely used to average elements of linear Euclidean spaces. Depending on the application, it is usually referred to as the average, the barycenter, or the center of mass. The use of the geometric mean, on the other hand, has been limited to positive numbers and positive integrable functions [13, 7]. In 1975, Anderson and Trapp [2] and Pusz and Woronowicz [22] introduced the harmonic and geometric means for a pair of positive operators on a Hilbert space. Thereafter, an extensive theory on operator means originated. It has been shown that the geometric mean of two positive-definite operators shares many of the properties of the geometric mean of two positive numbers. A recent paper by Lawson and Lim [16] surveys eight shared properties. The geometric mean of positive operators has been used mainly as a binary operation.

In [26], there was a discussion about how to define the geometric mean of more than two Hermitian semidefinite matrices. There have been attempts to use iterative procedures, but none seemed to work when the matrices do not commute. In [1] there is a definition for the geometric mean of a finite set of operators; however, the given definition is not invariant under reordering of the matrices. The present author, while working with means of a finite number of 3-dimensional rotation matrices [18], discovered that there is a close connection between the Riemannian mean of two rotations and the geometric mean of two Hermitian definite matrices. This observation motivated the present work on the generalization of the geometric mean for more than two matrices using metric-based means. In an abstract setting, if $\mathscr{M}$ is a Riemannian manifold with metric $\mathrm{d}(\cdot,\cdot)$, then by analogy to (1.1) and (1.2), a plausible definition of a mean associated with $\mathrm{d}(\cdot,\cdot)$ of m points in $\mathscr{M}$ is given by

$$\mathfrak{M}(x_1,\ldots,x_m) := \arg\min_{x\in\mathscr{M}} \sum_{k=1}^{m} \mathrm{d}(x_k,x)^2. \tag{1.3}$$

Note that this definition does not guarantee that the mean is unique.

As we have seen, for the set of positive real numbers, which is at the same time a Lie group and an open convex cone,[2] different notions of mean can be associated with different metrics. In what follows, we will extend these metric-based means to the cone of positive-definite transformations. The methods and ideas used in this paper carry over to the complex counterpart of the space considered here, i.e., the convex cone of Hermitian definite transformations. We here concentrate on the real space just for simplicity of exposition but not for any fundamental reason.

The remainder of this paper is organized as follows. In section 2 we gather all the necessary background from differential geometry and optimization on manifolds that will be used throughout the text. Further information on this condensed material can be found in [9, 5, 11, 25, 27]. In section 3 we give a Riemannian metric-based notion of mean for positive-definite matrices. We discuss some invariance properties of this mean and show that in the case where two matrices are to be averaged, this mean coincides with the geometric mean.

2. Preliminaries. Let $\mathcal{M}(n)$ be the set of $n\times n$ real matrices and $GL(n)$ be its subset containing only nonsingular matrices. $GL(n)$ is a Lie group, i.e., a group which is also a differentiable manifold and for which the operations of group multiplication

[2]Here and throughout we use the term *open convex cone*, or simply *cone*, when we really mean the interior of a convex cone.

and inverse are smooth. The tangent space at the identity is called the corresponding Lie algebra and denoted by $\mathfrak{gl}(n)$. It is the space of all linear transformations in $\mathbb{R}^n$, i.e., $\mathcal{M}(n)$.

In $\mathcal{M}(n)$ we shall use the Euclidean inner product, known as the Frobenius inner product and defined by $\langle \boldsymbol{A}, \boldsymbol{B} \rangle_F = \operatorname{tr}(\boldsymbol{A}^T \boldsymbol{B})$, where $\operatorname{tr}(\cdot)$ stands for the trace and the superscript ^{T} denotes the transpose. The associated norm $\|\boldsymbol{A}\|_F = \langle \boldsymbol{A}, \boldsymbol{A} \rangle_F^{1/2}$ is used to define the Euclidean distance on $\mathcal{M}(n)$,

$$\mathrm{d}_F(\boldsymbol{A}, \boldsymbol{B}) = \|\boldsymbol{A} - \boldsymbol{B}\|_F. \tag{2.1}$$

2.1. Exponential and logarithms. The exponential of a matrix in $\mathfrak{gl}(n)$ is given, as usual, by the convergent series

$$\exp \boldsymbol{A} = \sum_{k=0}^{\infty} \frac{1}{k!} \boldsymbol{A}^k. \tag{2.2}$$

We remark that the product of the exponentials of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is equal to $\exp(\boldsymbol{A} + \boldsymbol{B})$ only when $\boldsymbol{A}$ and $\boldsymbol{B}$ commute.

Logarithms of $\boldsymbol{A}$ in $GL(n)$ are solutions of the matrix equation $\exp \boldsymbol{X} = \boldsymbol{A}$. When $\boldsymbol{A}$ does not have eigenvalues in the (closed) negative real line, there exists a unique real logarithm, called the principal logarithm and denoted by $\operatorname{Log} \boldsymbol{A}$, whose spectrum lies in the infinite strip $\{z \in \mathbb{C} \ : \ -\pi < \operatorname{Im}(z) < \pi\}$ of the complex plane [9]. Furthermore, if for any given matrix norm $\|\cdot\|$ we have $\|\boldsymbol{I} - \boldsymbol{A}\| < 1$, where $\boldsymbol{I}$ denotes the identity transformation in $\mathbb{R}^n$, then the series $-\sum_{k=1}^{\infty} \frac{(\boldsymbol{I}-\boldsymbol{A})^k}{k}$ converges to $\operatorname{Log} \boldsymbol{A}$, and therefore one can write

$$\operatorname{Log} \boldsymbol{A} = -\sum_{k=1}^{\infty} \frac{(\boldsymbol{I} - \boldsymbol{A})^k}{k}. \tag{2.3}$$

We note that, in general, $\operatorname{Log}(\boldsymbol{AB}) \neq \operatorname{Log} \boldsymbol{A} + \operatorname{Log} \boldsymbol{B}$. We here recall the important fact [9]

$$\operatorname{Log}(\boldsymbol{A}^{-1} \boldsymbol{B} \boldsymbol{A}) = \boldsymbol{A}^{-1} (\operatorname{Log} \boldsymbol{B}) \boldsymbol{A}. \tag{2.4}$$

This fact is also true when Log in the above is replaced with an analytic matrix function.

The following result is essential in the development of our analysis.

PROPOSITION 2.1. *Let $\boldsymbol{X}(t)$ be a real matrix-valued function of the real variable t. We assume that, for all t in its domain, $\boldsymbol{X}(t)$ is an invertible matrix which does not have eigenvalues on the closed negative real line. Then*

$$\frac{d}{dt} \operatorname{tr} \left[\operatorname{Log}^2 \boldsymbol{X}(t) \right] = 2 \operatorname{tr} \left[\operatorname{Log} \boldsymbol{X}(t) \boldsymbol{X}^{-1}(t) \frac{d}{dt} \boldsymbol{X}(t) \right].$$

Proof. We recall the following facts:

(i) $\operatorname{tr}(\boldsymbol{AB}) = \operatorname{tr}(\boldsymbol{BA})$.

(ii) $\operatorname{tr}(\int_a^b \boldsymbol{M}(s) ds) = \int_a^b \operatorname{tr}(\boldsymbol{M}(s)) ds$.

(iii) $\operatorname{Log} \boldsymbol{A}$ commutes with $[(\boldsymbol{A} - \boldsymbol{I})s + \boldsymbol{I}]^{-1}$.

(iv) $\int_0^1 [(\boldsymbol{A} - \boldsymbol{I})s + \boldsymbol{I}]^{-2} \, ds = (\boldsymbol{I} - \boldsymbol{A})^{-1} \, [(\boldsymbol{A} - \boldsymbol{I})s + \boldsymbol{I}]^{-1} \, |_0^1 = \boldsymbol{A}^{-1}$.

(v) $\frac{d}{dt} \operatorname{Log} \boldsymbol{X}(t) = \int_0^1 [(\boldsymbol{X}(t) - \boldsymbol{I})s + \boldsymbol{I}]^{-1} \, \frac{d}{dt} \boldsymbol{X}(t) \, [(\boldsymbol{X}(t) - \boldsymbol{I})s + \boldsymbol{I}]^{-1} \, ds$.

Facts (i), (ii), (iii), and (iv) are easily checked. See [10] for a proof of (v).

Using the above, we have

$$
\begin{aligned}
\frac{d}{dt}\operatorname{tr}\left([\operatorname{Log}\boldsymbol{X}(t)]^2\right) &\overset{\text{(i)}}{=} 2\operatorname{tr}\left(\operatorname{Log}\boldsymbol{X}(t)\frac{d}{dt}\operatorname{Log}\boldsymbol{X}(t)\right)\\
&\overset{\text{(v)}}{=} 2\operatorname{tr}\left(\operatorname{Log}\boldsymbol{X}(t)\int_0^1\left[(\boldsymbol{X}(t)-\boldsymbol{I})s+\boldsymbol{I}\right]^{-1}\frac{d}{dt}\boldsymbol{X}(t)\left[(\boldsymbol{X}(t)-\boldsymbol{I})s+\boldsymbol{I}\right]^{-1}ds\right)\\
&= 2\operatorname{tr}\left(\int_0^1\operatorname{Log}\boldsymbol{X}(t)\left[(\boldsymbol{X}(t)-\boldsymbol{I})s+\boldsymbol{I}\right]^{-1}\frac{d}{dt}\boldsymbol{X}(t)\left[(\boldsymbol{X}(t)-\boldsymbol{I})s+I\right]^{-1}ds\right)\\
&\overset{\text{(ii)}}{=} 2\int_0^1\operatorname{tr}\left(\operatorname{Log}\boldsymbol{X}(t)\left[(\boldsymbol{X}(t)-\boldsymbol{I})s+\boldsymbol{I}\right]^{-1}\frac{d}{dt}\boldsymbol{X}(t)\left[(\boldsymbol{X}(t)-\boldsymbol{I})s+\boldsymbol{I}\right]^{-1}\right)ds\\
&\overset{\text{(i)}}{=} 2\int_0^1\operatorname{tr}\left(\left[(\boldsymbol{X}(t)-\boldsymbol{I})s+\boldsymbol{I}\right]^{-1}\operatorname{Log}\boldsymbol{X}(t)\left[(\boldsymbol{X}(t)-\boldsymbol{I})s+\boldsymbol{I}\right]^{-1}\frac{d}{dt}\boldsymbol{X}(t)\right)ds\\
&\overset{\text{(iii)}}{=} 2\int_0^1\operatorname{tr}\left(\operatorname{Log}\boldsymbol{X}(t)\left[(\boldsymbol{X}(t)-\boldsymbol{I})s+\boldsymbol{I}\right]^{-2}\frac{d}{dt}\boldsymbol{X}(t)\right)ds\\
&= 2\operatorname{tr}\left(\operatorname{Log}\boldsymbol{X}(t)\int_0^1\left[(\boldsymbol{X}(t)-\boldsymbol{I})s+I\right]^{-2}ds\frac{d}{dt}\boldsymbol{X}(t)\right)\\
&\overset{\text{(iv)}}{=} 2\operatorname{tr}\left(\operatorname{Log}\boldsymbol{X}(t)\boldsymbol{X}^{-1}(t)\frac{d}{dt}\boldsymbol{X}(t)\right). \qquad \square
\end{aligned}
$$

2.2. Gradient and geodesic convexity. For a real-valued function $f(x)$ defined on a Riemannian manifold $\mathscr{M}$, the gradient ∇f is the unique tangent vector u at x such that

$$\langle u, \nabla f\rangle = \left.\frac{d}{dt}f(\gamma(t))\right|_{t=0}, \tag{2.5}$$

where $\gamma(t)$ is a geodesic emanating from x in the direction of u, and $\langle\cdot,\cdot\rangle$ denotes the Riemannian inner product on the tangent space.

A subset $\mathscr{A}$ of a Riemannian manifold $\mathscr{M}$ is said to be convex if the shortest geodesic curve between any two points x and y in $\mathscr{A}$ is unique in $\mathscr{M}$ and lies in $\mathscr{A}$. A real-valued function defined on a convex subset $\mathscr{A}$ of $\mathscr{M}$ is said to be convex if its restriction to any geodesic path is convex, i.e., if $t \mapsto \hat{f}(t) \equiv f(\exp_x(tu))$ is convex over its domain for all $x \in \mathscr{M}$ and $u \in T_x(\mathscr{M})$, where $\exp_x$ is the exponential map at x.

2.3. The cone of the positive-definite symmetric matrices. We denote by

$$\mathcal{S}(n) = \{\boldsymbol{A} \in \mathcal{M}(n),\ \boldsymbol{A}^T = \boldsymbol{A}\}$$

the space of all $n \times n$ symmetric matrices and denote by

$$\mathcal{P}(n) = \{\boldsymbol{A} \in \mathcal{S}(n),\ \boldsymbol{A} > 0\}$$

the set of all $n \times n$ positive-definite symmetric matrices. Here $\boldsymbol{A} > 0$ means that the quadratic form $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} > 0$ for all $\boldsymbol{x} \in \mathbb{R}^n\backslash\{\boldsymbol{0}\}$. It is well known that $\mathcal{P}(n)$ is an open convex cone; i.e., if $\boldsymbol{P}$ and $\boldsymbol{Q}$ are in $\mathcal{P}(n)$, so is $\boldsymbol{P} + t\boldsymbol{Q}$ for any $t > 0$.

We recall that the exponential map from $\mathcal{S}(n)$ to $\mathcal{P}(n)$ is one-to-one and onto. In other words, the exponential of any symmetric matrix is a positive-definite symmetric matrix, and the inverse of the exponential (i.e., the principal logarithm) of any positive-definite symmetric matrix is a symmetric matrix.

As $\mathcal{P}(n)$ is an open subset of $\mathcal{S}(n)$, for each $\boldsymbol{P} \in \mathcal{P}(n)$ we identify the set $T_{\boldsymbol{P}}$ of tangent vectors to $\mathcal{P}(n)$ at $\boldsymbol{P}$ with $\mathcal{S}(n)$. On the tangent space at $\boldsymbol{P}$ we define the positive-definite inner product and corresponding norm,

$$\langle \boldsymbol{A}, \boldsymbol{B} \rangle_{\boldsymbol{P}} = \operatorname{tr}(\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}^{-1}\boldsymbol{B}), \qquad \|\boldsymbol{A}\|_{\boldsymbol{P}} = \langle \boldsymbol{A}, \boldsymbol{A} \rangle_{\boldsymbol{P}}^{1/2}, \tag{2.6}$$

that depend on the point $\boldsymbol{P}$. The positive definiteness is a consequence of the positive definiteness of the Frobenius inner product for

$$\langle \boldsymbol{A}, \boldsymbol{A} \rangle_{\boldsymbol{P}} = \operatorname{tr}(\boldsymbol{P}^{-1/2}\boldsymbol{A}\boldsymbol{P}^{-1/2}\boldsymbol{P}^{-1/2}\boldsymbol{A}\boldsymbol{P}^{-1/2}) = \left\langle \boldsymbol{P}^{-1/2}\boldsymbol{A}\boldsymbol{P}^{-1/2}, \boldsymbol{P}^{-1/2}\boldsymbol{A}\boldsymbol{P}^{-1/2} \right\rangle.$$

Let $[a, b]$ be a closed interval in $\mathbb{R}$, and let $\boldsymbol{\Gamma} : [a, b] \to \mathcal{P}(n)$ be a sufficiently smooth curve in $\mathcal{P}(n)$. We define the length of $\boldsymbol{\Gamma}$ by

$$\mathcal{L}(\boldsymbol{\Gamma}) := \int_a^b \sqrt{\left\langle \dot{\boldsymbol{\Gamma}}(t), \dot{\boldsymbol{\Gamma}}(t) \right\rangle_{\boldsymbol{\Gamma}(t)}}\, dt = \int_a^b \sqrt{\operatorname{tr}(\boldsymbol{\Gamma}(t)^{-1}\dot{\boldsymbol{\Gamma}}(t))^2}\, dt. \tag{2.7}$$

We note that the length $\mathcal{L}(\boldsymbol{\Gamma})$ is invariant under congruent transformations, i.e., $\boldsymbol{\Gamma} \mapsto \boldsymbol{C}\boldsymbol{\Gamma}\boldsymbol{C}^T$, where $\boldsymbol{C}$ is any fixed element of $\in GL(n)$. As $\frac{d}{dt}\boldsymbol{\Gamma}^{-1} = -\boldsymbol{\Gamma}^{-1}\dot{\boldsymbol{\Gamma}}\boldsymbol{\Gamma}^{-1}$, one can readily see that this length is also invariant under inversion.

The distance between two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ in $\mathcal{P}(n)$ considered as a differentiable manifold is the infimum of lengths of curves connecting them:

$$\mathrm{d}_{\mathcal{P}(n)}(\boldsymbol{A}, \boldsymbol{B}) := \inf \{ \mathcal{L}(\boldsymbol{\Gamma}) \mid \boldsymbol{\Gamma} : [a, b] \to \mathcal{P}(n) \text{ with } \boldsymbol{\Gamma}(a) = \boldsymbol{A},\ \boldsymbol{\Gamma}(b) = \boldsymbol{B} \}. \tag{2.8}$$

This metric makes $\mathcal{P}(n)$ a Riemannian manifold which is of dimension $\frac{1}{2}n(n+1)$. The geodesic emanating from $\boldsymbol{I}$ in the direction of $\boldsymbol{S}$, a (symmetric) matrix in the tangent space, is given explicitly by $e^{t\boldsymbol{S}}$ [17]. Using invariance under congruent transformations, the geodesic $\boldsymbol{P}(t)$ such that $\boldsymbol{P}(0) = \boldsymbol{P}$ and $\dot{\boldsymbol{P}}(0) = \boldsymbol{S}$ is therefore given by

$$\boldsymbol{P}(t) = \boldsymbol{P}^{1/2} e^{t\boldsymbol{P}^{-1/2}\boldsymbol{S}\boldsymbol{P}^{-1/2}} \boldsymbol{P}^{1/2}.$$

It follows that the Riemannian distance of $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ in $\mathcal{P}(n)$ is

$$\mathrm{d}_{\mathcal{P}(n)}(\boldsymbol{P}_1, \boldsymbol{P}_2) = \| \operatorname{Log}(\boldsymbol{P}_1^{-1}\boldsymbol{P}_2) \|_F = \left[\sum_{i=1}^{n} \ln^2 \lambda_i \right]^{1/2}, \tag{2.9}$$

where $\lambda_i,\ i = 1, \ldots, n$, are the eigenvalues of $\boldsymbol{P}_1^{-1}\boldsymbol{P}_2$. Even though in general $\boldsymbol{P}_1^{-1}\boldsymbol{P}_2$ is not symmetric, its eigenvalues are real and positive. This can be seen by noting that $\boldsymbol{P}_1^{-1}\boldsymbol{P}_2$ is similar to the positive-definite symmetric matrix $\boldsymbol{P}_2^{1/2}\boldsymbol{P}_1^{-1}\boldsymbol{P}_2^{1/2}$. It is important to note here that the real-valued function defined on $\mathcal{P}(n)$ by $\boldsymbol{P} \mapsto \mathrm{d}_{\mathcal{P}(n)}(\boldsymbol{P}, \boldsymbol{S})$, where $\boldsymbol{S} \in \mathcal{P}(n)$ is fixed, is (geodesically) convex [21].

We note in passing that $\mathcal{P}(n)$ is a homogeneous space of the Lie group $GL(n)$ (by identifying $\mathcal{P}(n)$ with the quotient $GL(n)/O(n)$). It is also a symmetric space of noncompact type [25].

We shall also consider the symmetric space of special positive matrices

$$\mathcal{SP}(n) = \{ \boldsymbol{A} \in \mathcal{P}(n),\ \det \boldsymbol{A} = 1 \}.$$

This submanifold can also be identified with the quotient $SL(n)/SO(n)$. Here $SL(n)$ denotes the special linear group of all determinant-one matrices in $GL(n)$. We note that $\mathcal{SP}(n)$ is a totally geodesic submanifold of $\mathcal{P}(n)$ [17]. Now since

$$\mathcal{P}(n) = \mathcal{SP}(n) \times \mathbb{R}^+,$$

$\mathcal{P}(n)$ can be seen as a foliated manifold whose codimension-one leaves are isomorphic to the hyperbolic space $\mathbb{H}^p$, where $p = \frac{1}{2}n(n+1) - 1$.

3. Means of positive-definite symmetric matrices. Using definition (1.3) with the two distance functions (2.1) and (2.9), we introduce the two different notions of mean in $\mathcal{P}(n)$.

DEFINITION 3.1. *The mean in the Euclidean sense, i.e., associated with the metric* (2.1)*, of m given positive-definite symmetric matrices $\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_m$ is defined as*

$$\mathfrak{A}(\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_m) := \operatorname*{arg\,min}_{\boldsymbol{P} \in \mathcal{P}(n)} \sum_{k=1}^{m} \|\boldsymbol{P}_k - \boldsymbol{P}\|_F^2. \tag{3.1}$$

DEFINITION 3.2. *The mean in the Riemannian sense, i.e., associated with the metric* (2.9)*, of m given positive-definite symmetric matrices $\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_m$ is defined as*

$$\mathfrak{G}(\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_m) := \operatorname*{arg\,min}_{\boldsymbol{P} \in \mathcal{P}(n)} \sum_{k=1}^{m} \|\operatorname{Log}(\boldsymbol{P}_k^{-1}\boldsymbol{P})\|_F^2. \tag{3.2}$$

Before we proceed further, we note that both means satisfy the following desirable properties:

P1. Invariance under reordering: For any permutation σ of the numbers $1, \ldots, m$, we have

$$\mathfrak{M}(\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_m) = \mathfrak{M}(\boldsymbol{P}_{\sigma(1)}, \boldsymbol{P}_{\sigma(2)}, \ldots, \boldsymbol{P}_{\sigma(m)}).$$

P2. Invariance under congruent transformations: If $\boldsymbol{P}$ is the positive-definite symmetric mean of $\{\boldsymbol{P}_k\}_{1 \le k \le m}$, then $\boldsymbol{C}\boldsymbol{P}\boldsymbol{C}^T$ is the positive-definite symmetric mean of $\{\boldsymbol{C}\boldsymbol{P}_k\boldsymbol{C}^T\}_{1 \le k \le m}$ for every $\boldsymbol{C}$ in $GL(n)$. From the special case when $\boldsymbol{C}$ is in the full orthogonal group $O(n)$, we deduce the invariance under orthogonal transformations.

We remark that P2 is the counterpart of the homogeneity property of means of positive numbers (but here left and right multiplication are both needed so that the resultant matrix lies in $\mathcal{P}(n)$). The mean in the Riemannian sense does satisfy the following additional property:

P3. Invariance under inversion: If $\boldsymbol{P}$ is the mean of $\{\boldsymbol{P}_k\}_{1 \le k \le m}$, then $\boldsymbol{P}^{-1}$ is the mean of $\{\boldsymbol{P}_k^{-1}\}_{1 \le k \le m}$.

The mean in the Euclidean sense does in fact satisfy properties other than P1 and P2; however, they are not relevant for the cone of positive-definite symmetric matrices. Furthermore, the solution of the minimization problem (3.1) is simply given by $\boldsymbol{P} = \frac{1}{m}\sum_{k=1}^{m} \boldsymbol{P}_k$, which is the usual arithmetic mean. Therefore, the mean in the Euclidean sense will not be considered any further.

Remark 3.3. The Riemannian mean of $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_m$ may also be called the *Riemannian barycenter* of $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_m$, which is a notion introduced by Grove, Karcher, and Ruh [12]. In [15] it was proven that for manifolds with nonpositive sectional curvature, the Riemannian barycenter is unique.

3.1. Characterization of the Riemannian mean. In the following proposition we will give a characterization of the Riemannian mean.

PROPOSITION 3.4. *The Riemannian mean of given m symmetric positive-definite matrices $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_m$ is the unique symmetric positive-definite solution to the nonlinear matrix equation*

$$\sum_{k=1}^{m} \operatorname{Log}(\boldsymbol{P}_k^{-1}\boldsymbol{P}) = \boldsymbol{0}. \tag{3.3}$$

Proof. First, we compute the derivative of the real-valued function $H(\boldsymbol{S}(t)) = \frac{1}{2}\| \operatorname{Log}(\boldsymbol{W}^{-1}\boldsymbol{S}(t))\|_F^2$ with respect to t, where $\boldsymbol{S}(t) = \boldsymbol{P}^{1/2}\exp(t\boldsymbol{A})\boldsymbol{P}^{1/2}$ is the geodesic emanating from $\boldsymbol{P}$ in the direction of $\boldsymbol{\Delta} = \dot{\boldsymbol{S}}(0) = \boldsymbol{P}^{1/2}\boldsymbol{A}\boldsymbol{P}^{1/2}$, and $\boldsymbol{W}$ is a constant matrix in $\mathcal{P}(n)$.

Using (2.4) and some properties of the trace, it follows that

$$H(\boldsymbol{S}(t)) = \frac{1}{2}\| \operatorname{Log}(\boldsymbol{W}^{-1/2}\boldsymbol{S}(t)\boldsymbol{W}^{-1/2})\|_F^2.$$

Because $\operatorname{Log}(\boldsymbol{W}^{-1/2}\boldsymbol{S}(t)\boldsymbol{W}^{-1/2})$ is symmetric, we have

$$\frac{d}{dt}H(\boldsymbol{S}(t))\bigg|_{t=0} = \frac{1}{2}\,\frac{d}{dt}\operatorname{tr}\left([\operatorname{Log}(\boldsymbol{W}^{-1/2}\boldsymbol{S}(t)\boldsymbol{W}^{-1/2})]^2\right)\bigg|_{t=0}.$$

Therefore, the general result of Proposition 2.1 applied to the above yields

$$\frac{d}{dt}H(\boldsymbol{S}(t))\bigg|_{t=0} = \operatorname{tr}[\operatorname{Log}(\boldsymbol{W}^{-1}\boldsymbol{P})\boldsymbol{P}^{-1}\boldsymbol{\Delta}] = \operatorname{tr}[\boldsymbol{\Delta}\operatorname{Log}(\boldsymbol{W}^{-1}\boldsymbol{P})\boldsymbol{P}^{-1}],$$

and hence the gradient of H is given by

$$\nabla H = \operatorname{Log}(\boldsymbol{W}^{-1}\boldsymbol{P})\boldsymbol{P}^{-1} = \boldsymbol{P}^{-1}\operatorname{Log}(\boldsymbol{P}\boldsymbol{W}^{-1}), \tag{3.4}$$

which is indeed in the tangent space, i.e., in $\mathcal{S}(n)$.

Now, let G denote the objective function of the minimization problem (3.2), i.e.,

$$G(\boldsymbol{P}) = \sum_{k=1}^{m} \| \operatorname{Log}(\boldsymbol{P}_k^{-1}\boldsymbol{P})\|_F^2. \tag{3.5}$$

Using the above, the gradient of G is found to be

$$\nabla G = \boldsymbol{P}\sum_{k=1}^{m} \operatorname{Log}(\boldsymbol{P}_k^{-1}\boldsymbol{P}). \tag{3.6}$$

As (3.5) is the sum of convex functions, the necessary condition and sufficient condition for $\boldsymbol{P}$ to be the minimum of (3.5) is the vanishing of the gradient (3.6), or, equivalently,

$$\sum_{k=1}^{m} \operatorname{Log}(\boldsymbol{P}_k^{-1}\boldsymbol{P}) = \boldsymbol{0}. \qquad \square$$

It is worth noting that the characterization for the Riemannian mean given in (3.3) is similar to the characterization

$$\sum_{k=1}^{m} \ln(x_k^{-1} x) = 0 \tag{3.7}$$

of the geometric mean (1.2) of positive numbers. However, unlike the case of positive numbers, where (3.7) yields to an explicit expression of the geometric mean, in general, due to the noncommutative nature of $\mathcal{P}(n)$, (3.3) cannot be solved in closed form. In the next section we will show that when $m = 2$, (3.3) yields explicit expressions of the Riemannian mean.

3.1.1. Riemannian mean of two positive-definite symmetric matrices. The following proposition shows that for the case $m = 2$, (3.3) can be solved analytically.

PROPOSITION 3.5. *The mean in the Riemannian sense of two positive-definite symmetric matrices $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ is given explicitly by the following six equivalent expressions:*

$$\begin{aligned} \mathfrak{G}(\boldsymbol{P}_1, \boldsymbol{P}_2) &= \boldsymbol{P}_1(\boldsymbol{P}_1^{-1}\boldsymbol{P}_2)^{1/2} = \boldsymbol{P}_2(\boldsymbol{P}_2^{-1}\boldsymbol{P}_1)^{1/2} \\ &= (\boldsymbol{P}_2\boldsymbol{P}_1^{-1})^{1/2}\boldsymbol{P}_1 = (\boldsymbol{P}_1\boldsymbol{P}_2^{-1})^{1/2}\boldsymbol{P}_2 \end{aligned} \tag{3.8}$$

$$\begin{aligned} &= \boldsymbol{P}_1^{1/2}(\boldsymbol{P}_1^{-1/2}\boldsymbol{P}_2\boldsymbol{P}_1^{-1/2})^{1/2}\boldsymbol{P}_1^{1/2} \\ &= \boldsymbol{P}_2^{1/2}(\boldsymbol{P}_2^{-1/2}\boldsymbol{P}_1\boldsymbol{P}_2^{-1/2})^{1/2}\boldsymbol{P}_2^{1/2}. \end{aligned} \tag{3.9}$$

Proof. First, we rewrite (3.3) as $\operatorname{Log}(\boldsymbol{P}_1^{-1}\boldsymbol{P}) = -\operatorname{Log}(\boldsymbol{P}_2^{-1}\boldsymbol{P})$. Then we take the exponential of both sides to obtain

$$\boldsymbol{P}_1^{-1}\boldsymbol{P} = \boldsymbol{P}^{-1}\boldsymbol{P}_2. \tag{3.10}$$

After left multiplying both sides with $\boldsymbol{P}_1^{-1}\boldsymbol{P}$ we get $(\boldsymbol{P}_1^{-1}\boldsymbol{P})^2 = \boldsymbol{P}_1^{-1}\boldsymbol{P}_2$. Such a matrix equation has $\boldsymbol{P}_1(\boldsymbol{P}_1^{-1}\boldsymbol{P}_2)^{1/2}$ as the unique solution in $\mathcal{P}(n)$. Therefore, the mean in the Riemannian sense of $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ is given explicitly by

$$\mathfrak{G}(\boldsymbol{P}_1, \boldsymbol{P}_2) = \boldsymbol{P}_1(\boldsymbol{P}_1^{-1}\boldsymbol{P}_2)^{1/2}.$$

The second equality in (3.9) can be easily verified by premultiplying $\boldsymbol{P}_1(\boldsymbol{P}_1^{-1}\boldsymbol{P}_2)^{1/2}$ by $\boldsymbol{P}_2\boldsymbol{P}_2^{-1} = \boldsymbol{I}$. This makes it clear that $\mathfrak{G}$ is symmetric with respect to $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, i.e., $\mathfrak{G}(\boldsymbol{P}_1, \boldsymbol{P}_2) = \mathfrak{G}(\boldsymbol{P}_2, \boldsymbol{P}_1)$. The third equality in (3.9) can be obtained by right multiplying both sides of (3.10) by $\boldsymbol{P}^{-1}\boldsymbol{P}_2$ and solving the resultant equation for $\boldsymbol{P}$. The fourth equality in (3.9) can be established from the third by right multiplying $(\boldsymbol{P}_2\boldsymbol{P}_1^{-1})^{1/2}\boldsymbol{P}_1$ by $\boldsymbol{P}_2^{-1}\boldsymbol{P}_2$. Alternatively, these equalities can be verified using (2.4). Furthermore, by use of (2.4) and after some algebra, one can show that the two expressions in (3.10) are but two other alternative forms of the geometric mean of $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ that highlight not only its symmetry with respect to $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ but also its symmetry as a matrix. ∎

The explicit equivalent expressions given in (3.9) and (3.10) for the Riemannian mean of a pair of positive-definite matrices that we obtained by solving the minimization problem (3.5) coincide with the different definitions of the geometric mean of a pair of positive Hermitian operators first introduced by Pusz and Woronowicz [22]. This mean arises in electrical network theory as described in the survey paper

by Trapp [26]. For this reason and the following proposition, the Riemannian mean will be termed the *geometric mean*.

PROPOSITION 3.6. *If all matrices* $\boldsymbol{P}_k$, $k = 1, \dots, m$, *belong to a single geodesic curve of* $\mathcal{P}(n)$, *i.e.,*

$$\boldsymbol{P}_k = \boldsymbol{C} \exp(t_k \boldsymbol{S}) \boldsymbol{C}^T, \quad k = 1, \dots, m,$$

where $\boldsymbol{S} \in \mathcal{S}(n)$, $\boldsymbol{C} \in GL(n)$, *and* $t_k \in \mathbb{R}$, $k = 1, \dots, m$, *then their Riemannian mean is*

$$\boldsymbol{P} = \boldsymbol{C} \exp\left(\frac{1}{m} \sum_{k=1}^{m} t_k \boldsymbol{S}\right) \boldsymbol{C}^T.$$

In particular, when $\boldsymbol{C}$ *is orthogonal, i.e., such that* $\boldsymbol{C}^T \boldsymbol{C} = \boldsymbol{I}$, *we have*

$$\boldsymbol{P} = (\boldsymbol{P}_1 \cdots \boldsymbol{P}_m)^{1/m} = \boldsymbol{P}_1^{1/m} \cdots \boldsymbol{P}_m^{1/m}.$$

Proof. Straightforward computations show that the given expression for $\boldsymbol{P}$ does satisfy (3.3) characterizing the Riemannian mean. ∎

For more than two matrices, in general, it is not possible to obtain an explicit expression for their geometric mean. In the commutative case of the cone of positive numbers, the problem of finding the geometric mean of three positive numbers can be done by first finding the geometric mean of two numbers and then finding the weighted geometric mean of the latter (with weight 2/3) and the other number (with weight 1/3). This procedure does not depend on the ordering of the numbers. For the space of positive-definite matrices this procedure gives different positive-definite matrices, depending on the way we order the elements. Furthermore, in general, none of these matrices satisfy the characterization (3.3) of the geometric mean. This is due to the fact that the geodesic triangle, whose sides join the three given matrices, is not flat. (See the discussion at the end of section 3.3.)

Thus, it is only in some special cases that one expects to have an explicit formula for the geometric mean. In the following proposition we give an example in which we obtain a closed-form expression of the geometric mean.

PROPOSITION 3.7. *Let* $\boldsymbol{P}_1$, $\boldsymbol{P}_2$, *and* $\boldsymbol{P}_3$ *be matrices in* $\mathcal{P}(n)$ *such that* $\boldsymbol{P}_1 = r\boldsymbol{P}_2$ *for some* $r > 0$. *Then their geometric mean is given explicitly by*

$$\begin{aligned}\mathfrak{G}(\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3) &= r^{1/3} \boldsymbol{P}_3 (\boldsymbol{P}_3^{-1} \boldsymbol{P}_2)^{2/3} = r^{-1/3} \boldsymbol{P}_3 (\boldsymbol{P}_3^{-1} \boldsymbol{P}_1)^{2/3} \\ &= r^{1/3} \boldsymbol{P}_2 (\boldsymbol{P}_2^{-1} \boldsymbol{P}_3)^{1/3} = r^{-1/3} \boldsymbol{P}_1 (\boldsymbol{P}_1^{-1} \boldsymbol{P}_3)^{1/3}.\end{aligned}$$

This proposition can easily be checked by straightforward calculations.

3.2. Reduction to the space of special positive-definite symmetric matrices. In this section we will show that the problem of finding the geometric mean of positive-definite matrices can be reduced to that of finding the geometric mean of special positive-definite matrices.

LEMMA 3.8. *If* $\boldsymbol{P}$ *is the geometric mean of* m *positive-definite symmetric matrices* $\boldsymbol{P}_1, \dots, \boldsymbol{P}_m$, *then for any* m*-tuple* $(a_1, \dots, a_m)$ *in the positive orthant of* $\mathbb{R}^m$, *the positive-definite symmetric matrix* $\sqrt[m]{a_1 \cdots a_m}\boldsymbol{P}$ *is the geometric mean of* $a_1\boldsymbol{P}_1, \dots, a_m\boldsymbol{P}_m$.

Proof. We have

$$\operatorname{Log}\left[(a_k \boldsymbol{P}_k)^{-1} \sqrt[m]{a_1 \cdots a_m}\boldsymbol{P}\right] = \operatorname{Log}(\boldsymbol{P}_k^{-1}\boldsymbol{P}) + \left[\frac{\ln a_1 + \cdots + \ln a_m}{m} - \ln a_k\right] \boldsymbol{I}.$$

Therefore,

$$\sum_{k=1}^{m} \operatorname{Log}\left[(a_k \boldsymbol{P}_k)^{-1}\, \sqrt[m]{a_1 \cdots a_m}\boldsymbol{P}\right] = \sum_{k=1}^{m} \operatorname{Log}(\boldsymbol{P}_k^{-1}\boldsymbol{P}) = \boldsymbol{0},$$

and hence $\sqrt[m]{a_1 \cdots a_m}\boldsymbol{P}$ is the geometric mean of $a_1\boldsymbol{P}_1, \ldots, a_m\boldsymbol{P}_m$. ∎

LEMMA 3.9. *If $\boldsymbol{P}$ and $\boldsymbol{Q}$ are in $\mathcal{SP}(n)$, i.e., are positive-definite symmetric matrices of determinant one, then for any $\alpha > 0$ we have $\mathrm{d}_{\mathcal{P}(n)}(\boldsymbol{P}, \boldsymbol{Q}) \leq \mathrm{d}_{\mathcal{P}(n)}(\boldsymbol{P}, \alpha\boldsymbol{Q})$, and equality holds if and only if $\alpha = 1$.*

Proof. This lemma follows immediately from the fact that $\mathcal{SP}(n)$ is a totally geodesic submanifold of $\mathcal{P}(n)$. Here is an alternative proof. Let $0 < \lambda_i,\ i = 1, \ldots, n$, be the eigenvalues of $\boldsymbol{P}^{-1}\boldsymbol{Q}$. Then

$$\mathrm{d}^2_{\mathcal{P}(n)}(\boldsymbol{P}, \alpha\boldsymbol{Q}) = \sum_{i=1}^{n} \ln^2(\alpha\lambda_i) = \sum_{i=1}^{n} \ln^2 \lambda_i + 2\ln\alpha \sum_{i=1}^{n} \ln\lambda_i + n\ln^2\alpha.$$

But as $\boldsymbol{P}$ and $\boldsymbol{Q}$ have determinant one, it follows that $\sum_{i=1}^{n} \ln\lambda_i = 0$, and hence

$$\mathrm{d}^2_{\mathcal{P}(n)}(\boldsymbol{P}, \alpha\boldsymbol{Q}) = \mathrm{d}^2_{\mathcal{P}(n)}(\boldsymbol{P}, \boldsymbol{Q}) + n\ln^2\alpha.$$

Therefore $\mathrm{d}_{\mathcal{P}(n)}(\boldsymbol{P}, \boldsymbol{Q}) \leq \mathrm{d}_{\mathcal{P}(n)}(\boldsymbol{P}, \alpha\boldsymbol{Q})$, where the equality holds only when $\alpha = 1$. ∎

PROPOSITION 3.10. *Given m positive-definite symmetric matrices $\{\boldsymbol{P}_k\}_{1\leq k\leq m}$, set $\alpha_k = \sqrt[n]{\det \boldsymbol{P}_k}$ and $\widetilde{\boldsymbol{P}}_k = \boldsymbol{P}_k/\alpha_k$. Then the geometric mean of $\{\boldsymbol{P}_k\}_{1\leq k\leq m}$ is the geometric mean of $\{\alpha_k\}_{1\leq k\leq m}$ multiplied by the geometric mean of $\{\widetilde{\boldsymbol{P}}_k\}_{1\leq k\leq m}$, i.e.,*

$$\mathfrak{G}(\boldsymbol{P}_1, \ldots, \boldsymbol{P}_m) = \sqrt[n]{\alpha_1 \cdots \alpha_m}\,\mathfrak{G}(\widetilde{\boldsymbol{P}}_1, \ldots, \widetilde{\boldsymbol{P}}_m).$$

The proof of this proposition is given by the combination of the results of the two previous lemmas.

3.3. Geometric mean of 2×2 positive-definite matrices. We start with the following geometric characterization of the geometric mean of two positive-definite matrices of determinant one.

PROPOSITION 3.11. *The geometric mean of two positive-definite symmetric matrices $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ in $\mathcal{SP}(2)$ is given by*

$$\mathfrak{G}(\boldsymbol{P}_1, \boldsymbol{P}_2) = \frac{\boldsymbol{P}_1 + \boldsymbol{P}_2}{\sqrt{\det(\boldsymbol{P}_1 + \boldsymbol{P}_2)}}.$$

Proof. Let $\boldsymbol{X} = (\boldsymbol{P}_1^{-1}\boldsymbol{P}_2)^{1/2}$. Note that $\det \boldsymbol{X} = 1$ and that the two eigenvalues λ and $1/\lambda$ of $\boldsymbol{X}$ are positive. By the Cayley–Hamilton theorem we have

$$\boldsymbol{X}^2 - \operatorname{tr}(\boldsymbol{X})\boldsymbol{X} + \boldsymbol{I} = \boldsymbol{0},$$

which after premultiplication by $\boldsymbol{P}_1$ and rearrangement is written as

$$\operatorname{tr}(\boldsymbol{X})\boldsymbol{P}_1(\boldsymbol{P}_1^{-1}\boldsymbol{P}_2)^{1/2} = \boldsymbol{P}_1 + \boldsymbol{P}_2.$$

But $\operatorname{tr}\boldsymbol{X} = \lambda + 1/\lambda$ and $\det(\boldsymbol{P}_1 + \boldsymbol{P}_2) = \det \boldsymbol{P_1} \det(\boldsymbol{I} + \boldsymbol{X}^2) = (1+\lambda^2)(1+1/\lambda^2) = (\lambda + 1/\lambda)^2$. Therefore,

$$\boldsymbol{P}_1(\boldsymbol{P}_1^{-1}\boldsymbol{P}_2)^{1/2} = \frac{\boldsymbol{P}_1 + \boldsymbol{P}_2}{\sqrt{\det(\boldsymbol{P}_1 + \boldsymbol{P}_2)}}.$$ ∎

This proposition gives a nice geometric characterization of the geometric mean of two positive-definite matrices in $\mathcal{SP}(2)$. It is given by the intersection of $\mathcal{SP}(2)$ with the ray joining the arithmetic average $\frac{1}{2}(\boldsymbol{P}_1 + \boldsymbol{P}_2)$ and the apex of the cone, i.e., the zero matrix.

By Lemma 3.8 and the above proposition, we have the following result giving an alternative expression for the geometric mean of two matrices in $\mathcal{P}(2)$ that does not require the evaluation of a matrix square root.

COROLLARY 3.12. *The geometric mean of two positive-definite symmetric matrices $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ in $\mathcal{P}(2)$ is given by*

$$\mathfrak{G}(\boldsymbol{P}_1, \boldsymbol{P}_2) = \sqrt{\alpha_1 \alpha_2} \frac{\sqrt{\alpha_2}\boldsymbol{P}_1 + \sqrt{\alpha_1}\boldsymbol{P}_2}{\sqrt{\det(\sqrt{\alpha_2}\boldsymbol{P}_1 + \sqrt{\alpha_1}\boldsymbol{P}_2)}},$$

where α_1 and α_2 are the determinants of $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, respectively.

Unfortunately, this elegant characterization of the geometric mean for two matrices in $\mathcal{SP}(2)$ cannot be generalized for more than two matrices in $\mathcal{SP}(2)$ nor for two matrices in $\mathcal{SP}(n)$ with $n > 2$. Indeed, this characterization, in general, does not hold for the mentioned cases.

Let us now consider in some detail the geometry of the space of 2×2 positive-definite matrices

$$\mathcal{P}(2) = \left\{ \begin{bmatrix} a & c \\ c & b \end{bmatrix} \ : \ a > 0, \ b > 0, \text{ and } ab > c^2 \right\}.$$

Set $0 < u = \frac{a+b}{2}$, $v = \frac{a-b}{2}$. Then the condition $ab > c^2$ can be rewritten as

$$\sqrt{v^2 + c^2} < u,$$

and therefore $\mathcal{P}(2)$ parameterized by u, v, and c can be seen as an open convex second-order cone (ice cream cone or future-light cone). Furthermore, the determinant-one condition $ab - c^2 = 1$ can be formulated as $u^2 - (v^2 + c^2) = 1$, and hence by using the identity $\cosh^2 \alpha - \sinh^2 \alpha = 1$, a matrix $\boldsymbol{P} = \left[\begin{smallmatrix} a & c \\ c & b \end{smallmatrix}\right]$ in $\mathcal{SP}(2)$ can be parameterized by $(\alpha, \theta) \in \mathbb{R} \times [0, \pi)$ as

$$\boldsymbol{P} = \cosh \alpha \boldsymbol{I} + \cos \theta \sinh \alpha \boldsymbol{J}_1 + \sin \theta \sinh \alpha \boldsymbol{J}_2,$$

where

$$\boldsymbol{J}_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \boldsymbol{J}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \cosh \alpha = \frac{a+b}{2}, \qquad \tan \theta = \frac{2c}{a-b}.$$

Note that $(\boldsymbol{I}, \ \boldsymbol{J}_1, \ \boldsymbol{J}_2)$ is a basis for the space $\mathcal{S}(2)$ of 2×2 symmetric matrices. Thus, we see that $\mathcal{SP}(2)$ is isomorphic to the hyperboloid $\mathbb{H}^2$. In particular, geodesics on $\mathcal{SP}(2)$ correspond to geodesics on $\mathbb{H}^2$.

For more than two matrices in $\mathcal{SP}(2)$, in general, it is not possible to obtain their geometric mean in closed form. Nevertheless, using the isomorphism between $\mathcal{SP}(2)$ and $\mathbb{H}^2$, we can identify the geometric mean with the hyperbolic centroid of point masses in the hyperboloid. In particular, the geometric mean of three matrices in $\mathcal{SP}(2)$ corresponds to the (hyperbolic) center of the hyperbolic triangles associated with the given three matrices. This center is the point of intersection of the three medians; see Figure 1. However, unlike Euclidean geometry, the ratio of the geodesic length between a vertex and the center to the length between this vertex and the midpoint on the opposite side is not 2/3 in general [6].

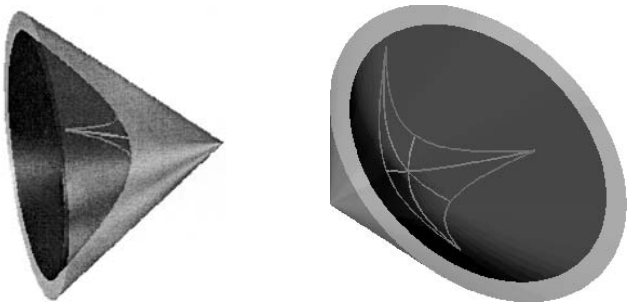


FIG. 1. *Two views for the representation, in the space parameterized by u, v, and c, of the cone $\mathcal{P}(2)$ and the hyperboloid $\mathcal{SP}(2)$. For the geodesic triangle shown, the point of concurrence of the three medians corresponds to the geometric mean of the positive-definite symmetric matrices in $\mathcal{P}(2)$ associated with the vertices of this triangle.*

4. Conclusion. Using the Riemannian metric on the space of positive-definite matrices, we defined the geometric mean. This mean satisfies some invariance properties. Some of these properties are related to the geodesic reversing isometry in the symmetric space considered here. Therefore, the notion of geometric mean, studied here and which is based on the Riemannian metric, can be used to define the geometric mean on other symmetric spaces which enjoy similar invariance properties. Here we used the space of positive-definite matrices as a prototype of a symmetric spaces of noncompact type. The case of the geometric mean of matrices in the group of special orthogonal matrices, which was studied in [18], is a prototype of symmetric spaces of compact type.

Equation (3.3) characterizing this mean is similar to (3.7) characterizing the geometric mean of positive numbers. Unfortunately, due to the noncommutative nature of the matrix multiplication, in general, it is not possible to obtain the geometric mean in closed form for more than two matrices.

Applications of the geometric mean to the problem of averaging data of anisotropic symmetric positive-definite tensors, such as in elasticity theory [8] and in diffusion tensor magnetic resonance imaging [3], are discussed in [19, 20, 4]. In [19], further invariance properties of the geometric mean are discussed and a fixed-point algorithm for solving the nonlinear matrix equation for the geometric mean of more than two matrices is presented.

REFERENCES

[1] M. ALIĆ, B. MOND, J. PEČARIĆ, AND V. VOLENEC, *Bounds for the differences of matrix means*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 119–123.

[2] W. N. ANDERSON, JR., AND G. E. TRAPP, *Shorted operators.* II, SIAM J. Appl. Math., 28 (1975), pp. 60–71.

[3] P. J. BASSER, J. MATIELLO, AND D. LE BIHAN, *MR diffusion tensor spectroscopy and imaging*, Biophys. J., 66 (1994), pp. 259–267.

[4] P. G. BATCHELOR, M. MOAKHER, D. ATKINSON, F. CALAMANTE, AND A. CONNELLY, *A rigorous framework for diffusion tensor calculus*, Magn. Reson. Med., 53 (2005), pp. 221–225.

[5] M. BERGER AND B. GOSTIAUX, *Differential Geometry: Manifolds, Curves, and Surfaces*, Springer-Verlag, New York, 1988.

[6] O. BOTTEMA, *On the medians of a triangle in hyperbolic geometry*, Canad. J. Math., 10 (1958), pp. 502–506.

[7] P. S. BULLEN, D. S. MITRINOVIĆ, AND P. M. VASIĆ, *Means and Their Inequalities*, Mathematics and Its Applications (East European Series) 31, D. Reidel Publishing Co., Dordrecht, The Netherlands, 1988.

[8] S. C. Cowin and G. Yang, *Averaging anisotropic elastic constant data*, J. Elasticity, 46 (1997), pp. 151–180.
[9] M. L. Curtis, *Matrix Groups*, Springer-Verlag, New York, Heidelberg, 1979.
[10] L. Dieci, B. Morini, and A. Papini, *Computational techniques for real logarithms of matrices*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 570–593.
[11] P. B. Eblerlein, *Goemetry of Nonpositively Curved Manifolds*, The University of Chicago Press, Chicago, 1996.
[12] K. Grove, H. Karcher, and E. A. Ruh, *Jacobi fields and Finsler metrics on compact Lie groups with an application to differentiable pinching problems*, Math. Ann., 211 (1974), pp. 7–21.
[13] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities*, Cambridge University Press, Cambridge, UK, 1934.
[14] T. Heath, *A History of Greek Mathematics, Vol.* 1: *From Thales to Euclid*, Dover, New York, 1981.
[15] H. Karcher, *Riemannian center of mass and mollifier smoothing*, Comm. Pure Appl. Math., 30 (1977), pp. 509–541.
[16] J. D. Lawson and Y. Lim, *The geometric mean, matrices, metrics, and more*, Amer. Math. Monthly, 108 (2001), pp. 797–812.
[17] H. Maass, *Siegel's Modular Forms and Dirichlet Series*, Lecture Notes in Math. 216, Springer-Verlag, Heidelberg, 1971.
[18] M. Moakher, *Means and averaging in the group of rotations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 1–16.
[19] M. Moakher, *On averaging symmetric positive-definite tensors*, J. Elasticity, submitted.
[20] 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, to appear.
[21] G. D. Mostow, *Strong Rigidity of Locally Symmetric Spaces*, Ann. Math. Stud. 78, Princeton University Press, Princeton, NJ, 1973.
[22] W. Pusz and S. L. Woronowicz, *Functional calculus for sesquilinear forms and the purification map*, Rep. Math. Phys., 8 (1975), pp. 159–170.
[23] M. Spiesser, *Les médiétés dans la pensée grecque*, in Musique et Mathématiques, Sci. Tech. Perspect. 23, Université de Nantes, Nantes, France, 1993, pp. 1–71.
[24] S. Stahl, *The Poincaré Half-Plane. A Gateway to Modern Geometry*, Jones and Bartlett, Boston, 1993.
[25] A. Terras, *Harmonic Analysis on Symmetric Spaces and Applications* II, Springer-Verlag, New York, 1988.
[26] G. E. Trapp, *Hermitian semidefinite matrix means and related matrix inequalities—an introduction*, Linear and Multilinear Algebra, 16 (1984), pp. 113–123.
[27] C. Udrişte, *Convex Functions and Optimization Methods on Riemannian Manifolds*, Math. Appl. 297, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 3, pp. 748–757

CHARACTERIZATION AND PROPERTIES OF (R,S)-SYMMETRIC, (R,S)-SKEW SYMMETRIC, AND (R,S)-CONJUGATE MATRICES*

WILLIAM F. TRENCH†

Abstract. Let $R \in \mathbb{C}^{m\times m}$ and $S \in \mathbb{C}^{n\times n}$ be nontrivial involutions; i.e., $R = R^{-1} \neq \pm I_m$ and $S = S^{-1} \neq \pm I_n$. We say that $A \in \mathbb{C}^{m\times n}$ is (R,S)-symmetric ((R,S)-skew symmetric) if $RAS = A$ ($RAS = -A$).

We give an explicit representation of an arbitrary (R,S)-symmetric matrix A in terms of matrices P and Q associated with R and U and V associated with S. If $R = R^*$, then the least squares problem for A can be solved by solving the independent least squares problems for $A_{PU} = P^*AU \in \mathbb{C}^{r\times k}$ and $A_{QV} = Q^*AV \in \mathbb{C}^{s\times \ell}$, where $r + s = m$ and $k + \ell = n$. If, in addition, either $\operatorname{rank}(A) = n$ or $S^* = S$, then $A^\dagger$ can be expressed in terms of $A_{PU}^\dagger$ and $A_{QV}^\dagger$. If $R = R^*$ and $S = S^*$, then a singular value decomposition of A can obtained from singular value decompositions of A_{PU} and A_{QV}. Similar results hold for (R,S)-skew symmetric matrices.

We say that $A \in \mathbb{C}^{m\times n}$ is R-conjugate if $RAS = \overline{R}$, where $R \in \mathbb{R}^{m\times m}$ and $S \in \mathbb{R}^{n\times n}$, $R = R^{-1} \neq \pm I_m$, and $S = S^{-1} \neq \pm I_n$. In this case $\Re(A)$ is (R,S)-symmetric and $\Im(A)$ is (R,S)-skew symmetric, so our results provide explicit representations for (R,S)-conjugate matrices. If $R^T = R$, then the least squares problem for the complex matrix A reduces to two least squares problems for a real matrix K. If, in addition, either $\operatorname{rank}(A) = n$ or $S^T = S$, then $A^\dagger$ can be obtained from $K^\dagger$. If both $R^T = R$ and $S^T = S$, a singular value decomposition of A can be obtained from a singular value decomposition of K.

Key words. least squares, Moore–Penrose inverse, optimal solution, (R,S)-conjugate, (R,S)-skew symmetric, (R,S)-symmetric

AMS subject classifications. 15A18, 15A57

DOI. 10.1137/S089547980343134X

1. Introduction.

In this paper we expand on a problem initiated by Chen [1], who considered matrices $A \in \mathbb{C}^{m\times n}$ such that

$$RAS = A \quad \text{or} \quad RAS = -A, \tag{1.1}$$

where $R \in \mathbb{C}^{m\times m}$ and $S \in \mathbb{C}^{n\times n}$ are involutory Hermitian matrices; i.e., $R = R^*$, $R^2 = I_m$, $S = S^*$, and $S^2 = I_n$. Chen cited applications involving such matrices, developed some of their theoretical properties, and indicated with numerical examples that the least squares problem for a matrix of rank n with either property reduces to two independent least squares problems for matrices of smaller dimensions. He also considered properties of the Moore–Penrose inverses of such matrices but did not obtain explicit expressions for them in terms of Moore–Penrose inverses of lower order matrices.

Here we characterize the matrices $A \in \mathbb{C}^{m\times n}$ satisfying (1.1) without assuming that R and S are Hermitian. We obtain general results on the least squares problem for the case where R is Hermitian, without assuming that S is Hermitian or that $\operatorname{rank}(A) = n$. Under the additional assumption that either S is Hermitian or $\operatorname{rank}(A) = n$, we obtain explicit expressions for $A^\dagger$ in terms of the Moore–Penrose inverses of two related matrices with smaller dimensions. Finally, under the assumption

*Received by the editors July 6, 2003; accepted for publication (in revised form) by D. Calvetti June 11, 2004; published electronically April 8, 2005.
http://www.siam.org/journals/simax/26-3/43134.html
†Trinity University, San Antonio, TX (wtrench@trinity.edu). Mailing address: 95 Pine Lane, Woodland Park, CO 80863.

that $R = R^*$ and $S = S^*$, we obtain a singular value decomposition of A in terms of singular value decompositions of these related matrices.

Under the assumption that $R \in \mathbb{R}^{m\times m}$ and $S \in \mathbb{R}^{n\times n}$, we consider the analogous questions for matrices $A \in \mathbb{C}^{m\times n}$ such that $RAS = \overline{A}$, so that $R\Re(A)S = \Re(A)$ and $R\Im(A)S = -\Im(A)$. We say that such matrices are (R,S)-conjugate.

We gave related results for square matrices with $R = S$ in [5] and studied other approximation problems for (R,S)-symmetric and (R,S)-skew symmetric matrices in [6].

2. Preliminary considerations. Let $R \in \mathbb{C}^{m\times m}$ and $S \in \mathbb{C}^{n\times n}$ be nontrivial involutions; thus $R = R^{-1} \neq \pm I_m$ and $S = S^{-1} \neq \pm I_n$. Then the minimal and characteristic polynomials of R are

$$m_R(x) = (x-1)(x+1) \quad \text{and} \quad c_R(x) = (x-1)^r(x+1)^s,$$

where $1 \leq r, s \leq m$ and $r + s = m$. Therefore there are matrices $P \in \mathbb{C}^{m\times r}$ and $Q \in \mathbb{C}^{m\times s}$ such that

$$P^*P = I_r, \quad Q^*Q = I_s, \tag{2.1}$$

$$RP = P, \quad \text{and} \quad RQ = -Q. \tag{2.2}$$

Thus, the columns of P (Q) form an orthonormal basis for the eigenspace of R associated with the eigenvalue $\lambda = 1$ ($\lambda = -1$). Although P and Q are not unique, suitable P and Q can be obtained by applying the Gram–Schmidt procedure to the columns of $I + R$ and $I - R$, respectively. If R is a signed permutation matrix, this requires little computation. For example, if J is the flip matrix with ones on the secondary diagonal and zeros elsewhere and $R = J_{2k}$, we can take

$$P = \frac{1}{\sqrt{2}}\begin{bmatrix} I_k \\ J_k \end{bmatrix} \quad \text{and} \quad Q = \frac{1}{\sqrt{2}}\begin{bmatrix} I_k \\ -J_k \end{bmatrix},$$

while if $R = J_{2k+1}$, we can take

$$P = \frac{1}{\sqrt{2}}\begin{bmatrix} I_k & 0_{k\times 1} \\ 0_{1\times k} & \sqrt{2} \\ J_k & 0_{k\times 1} \end{bmatrix} \quad \text{and} \quad Q = \frac{1}{\sqrt{2}}\begin{bmatrix} I_k \\ 0_{1\times k} \\ -J_k \end{bmatrix}.$$

If we define

$$\widehat{P} = \frac{P^*(I+R)}{2} \quad \text{and} \quad \widehat{Q} = \frac{Q^*(I-R)}{2}, \tag{2.3}$$

then

$$\widehat{P}P = I_r, \quad \widehat{P}Q = 0 \quad \widehat{Q}P = 0, \quad \text{and} \quad \widehat{Q}Q = I_s,$$

so

$$[P\ Q]^{-1} = \begin{bmatrix} \widehat{P} \\ \widehat{Q} \end{bmatrix}. \tag{2.4}$$

Similarly, there are integers k and ℓ such that $k+\ell=n$ and matrices $U\in\mathbb{C}^{n\times k}$ and $V\in\mathbb{C}^{n\times\ell}$ such that

$$U^*U=I_k,\quad V^*V=I_\ell,$$

$$SU=U,\quad\text{and}\quad SV=-V. \tag{2.5}$$

Moreover, if we define

$$\widehat{U}=\frac{U^*(I+S)}{2}\quad\text{and}\quad\widehat{V}=\frac{V^*(I-S)}{2}, \tag{2.6}$$

then

$$\widehat{U}U=I_k,\quad\widehat{U}V=0,\quad\widehat{V}U=0,\quad\text{and}\quad\widehat{V}V=I_\ell, \tag{2.7}$$

so

$$[U\ V]^{-1}=\begin{bmatrix}\widehat{U}\\ \widehat{V}\end{bmatrix}. \tag{2.8}$$

It is straightforward to verify that if $R=R^*$, then $[P\ Q]$ and $[P\ iQ]$ are both unitary. Similarly, if $S=S^*$, then $[U\ V]$ and $[U\ iV]$ are both unitary. We will use this observation in several places without restating it.

From (2.4) and (2.8), any $A\in\mathbb{C}^{m\times n}$ can be written conformably in block form as

$$A=[P\ Q]\begin{bmatrix}A_{PU}&A_{PV}\\ A_{QU}&A_{QV}\end{bmatrix}\begin{bmatrix}\widehat{U}\\ \widehat{V}\end{bmatrix}. \tag{2.9}$$

We say that $A\in\mathbb{C}^{m\times n}$ is (R,S)-symmetric if $RAS=A$, or (R,S)-skew symmetric if $RAS=-A$. From (2.2), (2.5), and (2.6),

$$RAS=[P\ Q]\begin{bmatrix}A_{PU}&-A_{PV}\\ -A_{QU}&A_{QV}\end{bmatrix}\begin{bmatrix}\widehat{U}\\ \widehat{V}\end{bmatrix}. \tag{2.10}$$

Henceforth, $z\in\mathbb{C}^n$, $x\in\mathbb{C}^k$, $y\in\mathbb{C}^\ell$, $w\in\mathbb{C}^m$, $\phi\in\mathbb{C}^r$, and $\psi\in\mathbb{C}^s$. We say that w is R-symmetric (R-skew symmetric) if $Rw=w$ ($Rw=-w$). An arbitrary w can be written uniquely as $w=P\phi+Q\psi$ with $\phi=\widehat{P}w$ and $\psi=\widehat{Q}w$. From (2.2), $P\phi$ is R-symmetric and $Q\psi$ is R-skew symmetric. Similarly, we say that z is S-symmetric (S-skew symmetric) if $Sz=z$ ($Sz=-z$). An arbitrary z can be written uniquely as $z=Ux+Vy$ with $x=\widehat{U}z$ and $y=\widehat{V}z$. From (2.5), Ux is S-symmetric and Vy is S-skew symmetric.

3. Two useful lemmas. Suppose that $B\in\mathbb{C}^{m\times n}$ and consider the least squares problem for B: If $w\in\mathbb{C}^m$, find $z\in\mathbb{C}^n$ such that

$$\|Bz-w\|=\min_{\zeta\in\mathbb{C}^n}\|B\zeta-w\|, \tag{3.1}$$

where $\|\cdot\|$ is the 2-norm. This problem has a unique solution if and only if $\operatorname{rank}(B)=n$. In this case, $z=(B^*B)^{-1}B^*w$. In any case, the optimal solution of (3.1) is the

unique n-vector z_0 of minimum norm that satisfies (3.1); thus, $z_0 = B^\dagger w$ where $B^\dagger$ is the Moore–Penrose inverse of B. The general solution of (3.1) is $z = z_0 + q$ with q in the null space of B, and

$$\|Bz - w\| = \|(BB^\dagger - I)w\|$$

for all such z.

The proof of the next lemma is motivated in part by a theorem of Meyer and Painter [3].

LEMMA 3.1. *Suppose that*

$$B = CKF, \tag{3.2}$$

where $C \in \mathbb{C}^{m\times m}$ is unitary and $F \in \mathbb{C}^{n\times n}$ is invertible. Then the general solution of (3.1) *is*

$$z = F^{-1}K^\dagger C^* w + (I - F^{-1}K^\dagger KF)h, \quad h \in \mathbb{C}^n, \tag{3.3}$$

and

$$\|Bz - w\| = \|(KK^\dagger - I)C^* w\| \tag{3.4}$$

for all such z. If either $\operatorname{rank}(B) = n$ *or F is unitary, then*

$$B^\dagger = F^{-1}K^\dagger C^*,$$

so the optimal solution of (3.1) *is*

$$z_0 = F^{-1}K^\dagger C^* w. \tag{3.5}$$

Moreover, z_0 is the unique solution of (3.1) *if* $\operatorname{rank}(B) = n$.

Proof. Recall [4] that $Z = W^\dagger$ and $W = Z^\dagger$ if and only if Z and W satisfy the Penrose conditions

$$WZW = W, \quad ZWZ = Z, \quad (ZW)^* = ZW, \quad \text{and} \quad (WZ)^* = WZ. \tag{3.6}$$

Let

$$B^L = F^{-1}K^\dagger C^*. \tag{3.7}$$

By letting $W = K$ and $Z = K^\dagger$ in (3.6), it is straightforward to verify that

$$B^L BB^L = B^L,\ BB^L B = B,\ (BB^L)^* = BB^L, \text{ and } B^L B = F^{-1}K^\dagger KF. \tag{3.8}$$

Any $\zeta \in \mathbb{C}^{n\times n}$ can be written as $\zeta = B^L w + q$, so

$$B\zeta - w = (BB^L - I)w + Bq.$$

From the second and third equalities in (3.8),

$$[(BB^L - I)w]^* Bq = 0,$$

so

$$\|B\zeta - w\|^2 = \|(BB^L - I)w\|^2 + \|Bq\|^2,$$

which is a minimum if and only if $Bq = 0$.

The second equality in (3.8) implies that $\operatorname{rank}(B^LB) = \operatorname{rank}(B)$, so $\operatorname{rank}(I-B^LB)$ equals the dimension of the null space of B. Now the second equality in (3.8) implies that $Bq = 0$ if and only if $q = (I - B^LB)h$, $h \in \mathbb{C}^{n\times n}$. Hence, the general solution of (3.1) is

$$z = B^Lw + (I - B^LB)h, \quad h \in \mathbb{C}^{n\times n}.$$

Substituting (3.2) and (3.7) into this yields (3.3). From (3.2) and (3.3),

$$Bz - w = C(KK^\dagger - I)C^*w,$$

since C is unitary. This implies (3.4).

If $\operatorname{rank}(B) = n$, then $\operatorname{rank}(K) = n$, so $K^\dagger K = I$ and the fourth equality in (3.8) reduces to $B^LB = I$. If F is unitary, the fourth equality in (3.8) implies that $(B^LB)^* = B^LB$. In either case, (3.8) implies that $B^L = B^\dagger$, so (3.5) is the optimal solution of (3.1). If $\operatorname{rank}(B) = n$, then (3.3) reduces to (3.5). ∎

The following lemma is obvious.

LEMMA 3.2. *Suppose that $B \in \mathbb{C}^{m\times n}$ and $B = CKF$, where $C \in \mathbb{C}^{m\times m}$ and $F \in \mathbb{C}^{n\times n}$ are unitary and $K = ZDW^*$ is a singular value decomposition of K. Then $B = (CZ)D(FW)^*$ is a singular value decomposition of B.*

4. Characterization and properties of (R, S)-symmetric matrices. The following theorem characterizes (R, S)-symmetric matrices.

THEOREM 4.1. *A is (R, S)-symmetric if and only if*

$$A = [P\ Q]\begin{bmatrix} A_{PU} & 0 \\ 0 & A_{QV} \end{bmatrix}\begin{bmatrix} \widehat{U} \\ \widehat{V} \end{bmatrix}, \tag{4.1}$$

where

$$A_{PU} = P^*AU \quad \text{and} \quad A_{QV} = Q^*AV. \tag{4.2}$$

Proof. From (2.9) and (2.10), $RAS = A$ if and only if (4.1) holds. If (4.1) holds, then (2.8) implies that

$$A[U\ V] = [P\ Q]\begin{bmatrix} A_{PU} & 0 \\ 0 & A_{QV} \end{bmatrix},$$

so $AU = PA_{PU}$ and $AV = QA_{QV}$. Therefore (2.1) implies (4.2).

The verification of the converse is straightforward. ∎

The following theorem reduces the least squares problem

$$\|Az - w\| = \min_{\zeta\in\mathbb{C}^n} \|A\zeta - w\| \tag{4.3}$$

to the independent $r \times k$ and $s \times \ell$ least squares problems

$$\|A_{PU}x - \phi\| = \min_{\xi\in\mathbb{C}^k} \|A_{PU}\xi - \phi\|$$

and

$$\|A_{QV}y - \psi\| = \min_{\eta\in\mathbb{C}^\ell} \|A_{QV}\eta - \psi\|.$$

THEOREM 4.2. *Suppose that A is (R,S)-symmetric, $R = R^*$, and $w = P\phi + Q\psi$. Then the general solution of* (4.3) *is*

$$z = U[A_{PU}^\dagger \phi + (I_k - A_{PU}^\dagger A_{PU})\xi] + V[A_{QV}^\dagger \psi + (I_\ell - A_{QV}^\dagger A_{QV})\eta],\ \xi \in \mathbb{C}^k,\ \eta \in \mathbb{C}^\ell,$$

and

$$\|Az - w\|^2 = \|(A_{PU}A_{PU}^\dagger - I_r)\phi\|^2 + \|(A_{QV}A_{QV}^\dagger - I_s)\psi\|^2$$

for all such z. If either $\operatorname{rank}(A) = n$ *or $S = S^*$, then*

$$A^\dagger = [U\ V] \begin{bmatrix} A_{PU}^\dagger & 0 \\ 0 & A_{QV}^\dagger \end{bmatrix} \begin{bmatrix} P^* \\ Q^* \end{bmatrix}$$

and $z_0 = UA_{PU}^\dagger \phi + VA_{QV}^\dagger \psi$ is the optimal solution of (4.3). *Moreover, z_0 is the unique solution of* (4.3) *if* $\operatorname{rank}(A) = n$.

Proof. Starting from Theorem 4.1, we apply Lemma 3.1 with

$$C = [P\ Q], \quad K = \begin{bmatrix} A_{PU} & 0 \\ 0 & A_{QV} \end{bmatrix}, \quad F = \begin{bmatrix} \widehat{U} \\ \widehat{V} \end{bmatrix},$$

$$z = Ux + Vy, \quad w = P\phi + Q\psi, \quad \text{and} \quad h = U\xi + V\eta.$$

It is straightforward to verify that

$$K^\dagger = \begin{bmatrix} A_{PU}^\dagger & 0 \\ 0 & A_{QV}^\dagger \end{bmatrix},$$

and the other details follow easily if we recall that since $R = R^*$, $\widehat{P} = P^*$ and $\widehat{Q} = Q^*$. ∎

Theorem 4.1 and Lemma 3.2 imply the following theorem. (Recall that $\widehat{U} = U^*$ and $\widehat{V} = V^*$ if $S = S^*$.)

THEOREM 4.3. *Suppose that $R = R^*$, $S = S^*$, and A is (R,S)-symmetric. Let*

$$A_{PU} = \Phi D_{PU} X^* \quad \textit{and} \quad A_{QV} = \Psi D_{QV} Y^*$$

be singular value decompositions of A_{PU} and A_{QV}. Then

$$A = [P\Phi\ Q\Psi] \begin{bmatrix} D_{PU} & 0 \\ 0 & D_{QV} \end{bmatrix} [UX\ VY]^*$$

is a singular value decomposition of A. Thus, the singular values of A_{PU} are singular values of A with associated R-symmetric left singular vectors and S-symmetric right singular vectors, and the singular values of A_{QV} are singular values of A with associated R-skew symmetric left singular vectors and S-skew symmetric right singular vectors.

5. Characterization and properties of (R,S)-skew symmetric matrices. The following theorem characterizes (R,S)-skew symmetric matrices.

THEOREM 5.1. *A is (R,S)-skew symmetric if and only if*

$$A=[P\ Q]\begin{bmatrix}0 & A_{PV}\\ A_{QU} & 0\end{bmatrix}\begin{bmatrix}\widehat{U}\\ \widehat{V}\end{bmatrix}, \tag{5.1}$$

where

$$A_{PV}=P^*AV \quad \textit{and} \quad A_{QU}=Q^*AU. \tag{5.2}$$

Proof. From (2.9) and (2.10), $RAS=-A$ if and only if (5.1) holds. If (5.1) holds, then (2.8) implies that

$$A[U\ V]=[P\ Q]\begin{bmatrix}0 & A_{PV}\\ A_{QU} & 0\end{bmatrix},$$

so $AU=QA_{QU}$ and $AV=PA_{PV}$. Therefore (2.1) implies (5.2).

The verification of the converse is straightforward. □

Theorem 5.1 and Lemma 3.1 imply the following theorem, which reduces (4.3) to the independent $s\times k$ and $r\times\ell$ least squares problems

$$\|A_{QU}x-\psi\|=\min_{\xi\in\mathbb{C}^k}\|A_{QU}\xi-\psi\|$$

and

$$\|A_{PV}y-\phi\|=\min_{\eta\in\mathbb{C}^\ell}\|A_{PV}\eta-\phi\|.$$

The proof is similar to the proof of Theorem 4.2, noting that in this case

$$K=\begin{bmatrix}0 & A_{PV}\\ A_{QU} & 0\end{bmatrix} \quad \text{and} \quad K^\dagger=\begin{bmatrix}0 & A_{QU}^\dagger\\ A_{PV}^\dagger & 0\end{bmatrix}.$$

THEOREM 5.2. *Suppose that A is (R,S)-skew symmetric, $R^*=R$, and $w=P\phi+Q\psi$. Then the general solution of* (4.3) *is*

$$z=U[A_{QU}^\dagger\psi+(I_k-A_{QU}^\dagger A_{QU})\xi]+V[A_{PV}^\dagger\phi+(I_\ell-A_{PV}^\dagger A_{PV})\eta],\ \xi\in\mathbb{C}^k,\ \eta\in\mathbb{C}^\ell,$$

and

$$\|Az-w\|^2=\|(A_{QU}A_{QU}^\dagger-I_s)\psi\|^2+\|(A_{PV}A_{PV}^\dagger-I_r)\phi\|^2$$

for all such z. If either $\operatorname{rank}(A)=n$ *or $S=S^*$, then*

$$A^\dagger=[U\ V]\begin{bmatrix}0 & A_{QU}^\dagger\\ A_{PV}^\dagger & 0\end{bmatrix}\begin{bmatrix}P^*\\ Q^*\end{bmatrix}$$

and $z_0=UA_{QU}^\dagger\psi+VA_{PV}^\dagger\phi$ is the optimal solution of (4.3). *Moreover, z_0 is the unique solution of* (4.3) *if* $\operatorname{rank}(A)=n$.

Theorem 5.1 and Lemma 3.2 imply the following theorem.

THEOREM 5.3. *Suppose that $R = R^*$, $S = S^*$, and A is (R,S)-skew symmetric. Let*

$$A_{PV} = \Phi D_{PV} Y^* \quad \textit{and} \quad A_{QU} = \Psi D_{QU} X^*$$

be singular value decompositions of A_{PV} and A_{QU}. Then

$$A = [P\Phi\ Q\Psi] \begin{bmatrix} D_{PV} & 0 \\ 0 & D_{QU} \end{bmatrix} [VY\ UX]^*$$

is a singular value decomposition of A. Thus, the singular values of A_{PV} are singular values of A with R-symmetric left singular vectors and S-skew symmetric right singular vectors, and the singular values of A_{QU} are singular values of A with R-skew symmetric left singular vectors and S-symmetric right singular vectors.

6. Characterization and properties of (R,S)-conjugate matrices. In this section we impose the following standing assumption.

ASSUMPTION A. *$R \in \mathbb{R}^{m\times m}$, $S \in \mathbb{R}^{n\times n}$ $R^{-1} = R \neq \pm I_m$, $S^{-1} = S \neq \pm I_n$, $P \in \mathbb{R}^{m\times r}$, $Q \in \mathbb{R}^{m\times s}$, $U \in \mathbb{R}^{n\times k}$, and $V \in \mathbb{R}^{n\times \ell}$. Also, $A = B + iC$ with B, $C \in \mathbb{R}^{m\times n}$.*

Under this assumption, (2.3) reduces to

$$\widehat{P} = \frac{P^T(I+R)}{2} \quad \text{and} \quad \widehat{Q} = \frac{Q^T(I-R)}{2},$$

and (2.6) reduces to

$$\widehat{U} = \frac{U^T(I+S)}{2}, \quad \text{and} \quad \widehat{V} = \frac{V^T(I-S)}{2}.$$

Moreover, if $R = R^T$, then $\widehat{P} = P^T$, $\widehat{Q} = Q^T$, and $[P\ iQ]$ is unitary. Similarly, if $S = S^T$, then $\widehat{U} = U^T$, $\widehat{V} = V^T$, and $[U\ iV]$ is unitary.

We say that A is (R,S)-conjugate if $RAS = \overline{A}$. The following theorem characterizes the class of (R,S)-conjugate matrices.

THEOREM 6.1. *$A = B + iC$ is (R,S)-conjugate if and only if*

$$A = [P\ iQ] \begin{bmatrix} B_{PU} & -C_{PV} \\ C_{QU} & B_{QV} \end{bmatrix} \begin{bmatrix} \widehat{U} \\ -i\widehat{V} \end{bmatrix}, \tag{6.1}$$

where

$$B_{PU} = P^T B U, \quad B_{QV} = Q^T B V, \quad C_{PV} = P^T C V, \quad C_{QU} = Q^T C U. \tag{6.2}$$

Proof. If $RAS = \overline{A}$, then $RBS = B$ and $RCS = -C$. Therefore Theorem 4.1 implies that

$$B = [P\ Q] \begin{bmatrix} B_{PU} & 0 \\ 0 & B_{QV} \end{bmatrix} \begin{bmatrix} \widehat{U} \\ \widehat{V} \end{bmatrix}$$

with B_{PU} and B_{QV} as in (6.2) and Theorem 5.1 implies that

$$C = [P\ Q] \begin{bmatrix} 0 & C_{PV} \\ C_{QU} & 0 \end{bmatrix} \begin{bmatrix} \widehat{U} \\ \widehat{V} \end{bmatrix}$$

with C_{PV} and C_{QU} as in (6.2). Therefore

$$A = B + iC = [P\ Q] \begin{bmatrix} B_{PU} & iC_{PV} \\ iC_{QU} & B_{QV} \end{bmatrix} \begin{bmatrix} \widehat{U} \\ \widehat{V} \end{bmatrix},$$

which is equivalent to (6.1).

For the converse, if A satisfies (6.1) where the center matrix is in $\mathbb{R}^{m\times n}$, then $RAS = \overline{A}$. Moreover, $A = B + iC$ with

$$B = PB_{PU}\widehat{U} + QB_{QV}\widehat{V} \quad \text{and} \quad C = QC_{QU}\widehat{U} + PC_{PV}\widehat{V}.$$

Now we invoke (2.1) (with $^* = ^T$) and (2.7) to verify (6.2). □

Theorem 6.1 with $m = n$ and $R = S$ is related to a result of Ikramov [2]. See also [5, Theorem 19].

Henceforth

$$K = \begin{bmatrix} B_{PU} & -C_{PV} \\ C_{QU} & B_{QV} \end{bmatrix} \in \mathbb{R}^{m\times n}.$$

An arbitrary z can be written uniquely as $z = Ux + iVy$ with $x = \widehat{U}z$ and $y = -i\widehat{V}z$. An arbitrary w can be written uniquely as $w = P\phi + iQ\psi$ with $\phi = \widehat{P}w$ and $\psi = -i\widehat{Q}w$. For our present purposes it is useful to write x, y, ϕ, and ψ in terms of their real and imaginary parts; thus,

$$x = x_1 + ix_2, \quad x_1, x_2 \in \mathbb{R}^k, \quad y = y_1 + iy_2, \quad y_1, y_2 \in \mathbb{R}^\ell,$$

$$\phi = \phi_1 + i\phi_2, \quad \phi_1, \phi_2 \in \mathbb{R}^r, \quad \psi = \psi_1 + i\psi_2, \quad \psi_1, \psi_2 \in \mathbb{R}^s.$$

Theorem 6.1 and Lemma 3.1 imply the following theorem, which reduces (4.3) to two independent least squares problems for the real matrix K:

$$\left\| K \begin{bmatrix} x_j \\ y_j \end{bmatrix} - \begin{bmatrix} \phi_j \\ \psi_j \end{bmatrix} \right\| = \min_{\xi_j\in\mathbb{R}^k, \eta_j\in\mathbb{R}^\ell} \left\| K \begin{bmatrix} \xi_j \\ \eta_j \end{bmatrix} - \begin{bmatrix} \phi_j \\ \psi_j \end{bmatrix} \right\|, \quad j = 1, 2.$$

THEOREM 6.2. *Suppose that A is (R, S)-conjugate, $R^T = R$, and $w = P\phi + Q\psi$. Then the general solution of* (4.3) *is*

$$z = [U\ iV] \left(K^\dagger \begin{bmatrix} \psi \\ \phi \end{bmatrix} + (I - K^\dagger K) \begin{bmatrix} \xi \\ \eta \end{bmatrix} \right), \quad \xi \in \mathbb{C}^k, \quad \eta \in \mathbb{C}^\ell,$$

and

$$\|Az - w\|^2 = \left\| (KK^\dagger - I) \begin{bmatrix} \psi_1 \\ \phi_1 \end{bmatrix} \right\|^2 + \left\| (KK^\dagger - I) \begin{bmatrix} \psi_2 \\ \phi_2 \end{bmatrix} \right\|^2$$

for all such z. If either $\operatorname{rank}(A) = n$ *or $S = S^T$, then*

$$A^\dagger = [U\ iV]K^\dagger \begin{bmatrix} P^T \\ -iQ^T \end{bmatrix}$$

and

$$z_0 = [U\ iV]K^\dagger \begin{bmatrix} \psi \\ \phi \end{bmatrix}$$

is the optimal solution of (4.3). *Moreover,* z_0 *is the unique solution of* (4.3) *if* $\operatorname{rank}(A) = n$.

Finally, Lemma 3.2 and Theorem 6.1 imply the following theorem.

THEOREM 6.3. *Suppose* $R^T = R$, $S^T = S$, *and* A *is* (R,S)*-conjugate. Let* $K = WDZ^T$ *be a singular value decomposition of* K. *Then*

$$A = [P\ iQ]WD([U\ iV]Z)^*$$

is a singular value decomposition of A. *Therefore the left singular vectors of* A *can be written as* $w_j = P\phi_j + iQ\psi_j$, *with* $\phi_j \in \mathbb{R}^r$ *and* $\psi_j \in \mathbb{R}^s$, $1 \le j \le m$, *and the right singular vectors of* A *can be written as* $z_j = Ux_j + iVy_j$ *with* $x_j \in \mathbb{R}^k$ *and* $y_j \in \mathbb{R}^\ell$, $1 \le j \le n$.

REFERENCES

[1] H.-C. CHEN, *Generalized reflexive matrices: Special properties and applications*, SIAM J. Matrix Anal. Appl. 19 (1998), pp. 140–153.

[2] KH. D. IKRAMOV, *The use of block symmetries to solve algebraic eigenvalue problems*, USSR Comput. Math. Math. Physics, 30 (1990), pp. 9–16.

[3] C. D. MEYER AND R. J. PAINTER, *Note on a least squares inverse for a matrix*, J. Assoc. Comput. Mach., 17 (1970), pp. 110–112.

[4] R. PENROSE, *A generalized inverse for matrices*, Proc. Camb. Philos. Soc., 51 (1955), pp. 406–413.

[5] W. F. TRENCH, *Characterization and properties of matrices with generalized symmetry or skew symmetry*, Linear Algebra Appl., 377 (2004), pp. 207–218.

[6] W. F. TRENCH, *Minimization problems for* (R,S)*-symmetric and* (R,S)*-skew symmetric matrices*, Linear Algebra Appl., 389 (2004), pp. 23–31.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 3, pp. 758–764

ON MATRIX POLYNOMIALS WITH REAL ROOTS*

LEONID GURVITS† AND LEIBA RODMAN‡

Abstract. It is proved that the roots of combinations of matrix polynomials with real roots can be recast as eigenvalues of combinations of real symmetric matrices, under certain hypotheses. The proof is based on the recent solution of the Lax conjecture. Several applications and corollaries, in particular concerning hyperbolic matrix polynomials, are presented.

Key words. hyperbolic polynomials, matrix polynomials

AMS subject classifications. 47A56, 15A57

DOI. 10.1137/040606089

1. Main result. A polynomial is called *hyperbolic* if all its roots are real. The class of hyperbolic polynomials is a classical well-studied class (see, e.g., [16]). There are at least two useful ways to extend this notion to polynomials with complex $n \times n$ matrix coefficients, in short, matrix polynomials. One way is to require that the determinant has only real roots; the other way involves using the quadratic form given by the matrix polynomial. Thus, a monic (i.e., with leading coefficient I_n, the $n \times n$ identity matrix) matrix polynomial $L(z)$ of degree ℓ is said to be *hyperbolic* if for every nonzero $x \in \mathbb{C}^n$ (the n-dimensional vector space of columns with complex components) the polynomial equation

$$\langle L(z)x, x\rangle = 0 \tag{1.1}$$

has ℓ real roots (counted with multiplicities). We denote here by $\langle \cdot, \cdot \rangle$ the standard inner product in $\mathbb{C}^n$. An $n \times n$ monic matrix polynomial $L(z)$ of degree ℓ will be called *weakly hyperbolic* if $\det L(z) = 0$ has $n\ell$ real roots (multiplicities counted). Note that our terminology differs slightly from the terminology in some sources (for example, [14]). Clearly, every hyperbolic matrix polynomial is weakly hyperbolic, and the coefficients of every hyperbolic matrix polynomial are Hermitian matrices. See, e.g., [15, 1, 14, 10] for the theory and applications of hyperbolic matrix and operator polynomials.

In this note we prove the following theorem. It states that the roots of combinations of hyperbolic matrix polynomials can be recast as eigenvalues of combinations of real symmetric matrices, under certain hypotheses. We denote by $\mathbb{R}$ the field of real numbers.

THEOREM 1.1. *Let*

$$L(z) = \sum_{j=0}^{\ell} L_j z^j, \qquad M(z) = \sum_{j=0}^{\ell} M_j z^j, \qquad M_\ell = L_\ell = I,$$

*Received by the editors March 31, 2004; accepted for publication (in revised form) by R. Bhatia August 6, 2004; published electronically April 8, 2005.
http://www.siam.org/journals/simax/26-3/60608.html
†Los Alamos National Laboratory, Los Alamos, NM 87545 (gurvits@lanl.gov).
‡College of William and Mary, Department of Mathematics, P. O. Box 8795, Williamsburg, VA 23187-8795 (lxrodm@math.wm.edu). The research leading to this article was done while the author visited LANL, whose hospitality is gratefully acknowledged. The research of this author was partially supported by NSF grant DMS-9988579.

be two monic $n \times n$ matrix polynomials such that

$$\alpha L(z) + (1-\alpha)M(z) \quad \text{is weakly hyperbolic for every } \alpha \in \mathbb{R}. \tag{1.2}$$

Then there exist $n\ell \times n\ell$ real symmetric matrices A and B such that for every $\alpha \in \mathbb{R}$, the roots of $\det(\alpha L(z) + (1-\alpha)M(z))$, counted according to their multiplicities, coincide with the eigenvalues of $\alpha A + (1-\alpha)B$, also counted according to their multiplicities.

Conversely, if the roots of $\det(\alpha L(z)+(1-\alpha)M(z))$ coincide with the eigenvalues of $\alpha A + (1-\alpha)B$ (counted with multiplicities) for every $\alpha \in \mathbb{R}$, where A and B are fixed real symmetric $n\ell \times n\ell$ matrices, then (1.2) holds.

Proof. Let

$$C_L = \begin{pmatrix} 0 & I_n & 0 & \dots & 0 \\ 0 & 0 & I_n & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & I_n \\ -L_0 & -L_1 & -L_2 & \dots & -L_{\ell-1} \end{pmatrix}$$

and

$$C_M = \begin{pmatrix} 0 & I_n & 0 & \dots & 0 \\ 0 & 0 & I_n & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & I_n \\ -M_0 & -M_1 & -M_2 & \dots & -M_{\ell-1} \end{pmatrix}$$

be the companion matrices of $L(z)$ and of $M(z)$, respectively. Then $\alpha C_L + (1-\alpha)C_M$ is the companion matrix of $\alpha L(z) + (1-\alpha)M(z)$, and therefore the roots of $\det(\alpha L(z)+(1-\alpha)M(z))$ (counted with multiplicities) coincide with the eigenvalues of $\alpha C_L + (1-\alpha)C_M$ (also counted with multiplicities), for every $\alpha \in \mathbb{R}$. (This is a standard fact in the theory of matrix polynomials; see, for example, [5].) Thus, $\alpha C_L + (1-\alpha)C_M$ has $n\ell$ real eigenvalues for every $\alpha \in \mathbb{R}$. Consider the homogeneous polynomial of three real variables α, β, γ:

$$P(\alpha, \beta, \gamma) := \det(\alpha C_L + \beta C_M - \gamma I_{n\ell}).$$

If $\alpha + \beta \neq 0$, the polynomial $P(\alpha, \beta, \gamma)$ (as a polynomial of γ) has $n\ell$ real roots (counted with multiplicities). If $\alpha + \beta = 0$, then

$$P(\alpha, \beta, \gamma) = \pm\gamma^{n(\ell-1)} \cdot \det(\alpha L_{\ell-1} + \beta M_{\ell-1} - \gamma I_n)$$

also has $n\ell$ real roots. To see this, we will show that the $n \times n$ matrix $L_{\ell-1} - M_{\ell-1}$ has n real eigenvalues (counted with multiplicities). Indeed, for a fixed positive ε consider the monic matrix polynomial $(\varepsilon)^{-1}((1+\varepsilon)L(z) - M(z))$ and its companion matrix $C_{(\varepsilon)^{-1}((1+\varepsilon)L-M)}$. By (1.2), the eigenvalues of $C_{(\varepsilon)^{-1}((1+\varepsilon)L-M)}$ are all real, and therefore the eigenvalues of the matrix

$$X := \lim_{\varepsilon \to 0} \left(\varepsilon C_{(\varepsilon)^{-1}((1+\varepsilon)L-M)}\right)$$

are also all real. However, the top $n(\ell-1)$ rows of X are zeros, and the bottom right $n \times n$ corner of X coincides with $M_{\ell-1} - L_{\ell-1}$. So the matrix $L_{\ell-1} - M_{\ell-1}$ has

all eigenvalues real. Thus, $P(\alpha, \beta, \gamma)$ is hyperbolic in the direction of $(0, 0, 1)$, in the sense of the Lax conjecture; see [11, 13]. By the main result of [13] (the proof in [13] is based on [7, 17]), we have

$$P(\alpha, \beta, \gamma) = \det(\alpha A + \beta B - \gamma I)$$

for some real symmetric matrices A and B. The direct statement of the theorem follows.

To prove the converse statement, simply reverse the argument, taking into account that real symmetric matrices have all eigenvalues real. □

For further development of the theory of hyperbolic polynomials of several variables and many applications, in particular, mixed determinants, see [6].

2. Corollaries and applications. We start by recalling Obreschkoff's theorem (see [16, 3]), which will be needed in the proof of the next corollary: Two real scalar polynomials $f(z)$ and $g(z)$ of degrees ℓ and $\ell - 1$, respectively, have the property that $f(z) + tg(z)$ has ℓ real roots (counted with multiplicities) for every real t if and only if $f(z)$ and $g(z)$ have ℓ and $\ell - 1$ real roots, respectively, and the roots of $f(z)$ and of $g(z)$ interlace (the cases when $f(z)$ or $g(z)$ has multiple roots and/or when $f(z)$ and $g(z)$ have common roots are not excluded here).

A proof of Obreschkoff's theorem can be given using the approach of Theorem 1.1, as follows. Below, we formulate Obreschkoff's theorem in a slightly different but equivalent form; for simplicity, it will be assumed that $f(z)$ and $h(z)$ are relatively prime (the general case is easily reduced to this one by dividing $f(z)$ and $h(z)$ by their greatest common divisor).

PROPOSITION 2.1. *The following statements are equivalent for scalar monic relatively prime polynomials $f(z)$ and $h(z)$ of degree ℓ:*

(1) *The polynomials $\alpha f(z) + \beta h(z)$, $\alpha, \beta \in \mathbb{R}$, $\alpha^2 + \beta^2 \neq 0$, have all their roots real;*
(2) *The polynomials $\alpha f(z) + (1 - \alpha) h(z)$, $\alpha \in \mathbb{R}$, have all their roots real;*
(3) *$f(z)$ has all its roots real, and the quotient $h(z)/f(z)$ has the form*

$$\frac{h(z)}{f(z)} = 1 + \sum_{j=1}^{p} \frac{c_j}{z - \lambda_j}, \tag{2.1}$$

where $\lambda_j \in \mathbb{R}$, and the real numbers c_j are all of the same sign;
(4) *Both $f(z)$ and $h(z)$ have ℓ distinct real roots, and the roots of $f(z)$ and of $h(z)$ interlace.*

Proof. (1) clearly implies (2). The statements (3) and (4) are equivalent: Indeed, if (3) or (4) holds true, then $f(z)$ has necessarily simple roots, and denoting the roots of $f(z)$ by $\lambda_1 < \cdots < \lambda_p$, we see that in the representation (2.1),

$$\operatorname{sign}(c_j) = \operatorname{sign} \frac{h(z)}{f(z)} \qquad \text{as } z \to \lambda_j,\ z > \lambda_j,$$

whereas

$$-\operatorname{sign}(c_{j+1}) = \operatorname{sign} \frac{h(z)}{f(z)} \qquad \text{as } z \to \lambda_{j+1},\ z < \lambda_{j+1}.$$

These equalities imply that that the roots of $f(z)$ and $h(z)$ interlace if and only if all the c_j's are of the same sign.

We next prove that (2) implies (3). Arguing as in the proof of Theorem 1.1, we obtain that the characteristic polynomial of $\alpha C_f + \beta C_h$ coincides with the characteristic polynomial of $\alpha A + \beta B$, for all $\alpha, \beta \in \mathbb{R}$, where A and B are fixed (independent of α and β) distinct real symmetric matrices. Taking $\alpha - \beta = 0$ we see that $\operatorname{rank}(A - B) \leq 1$. Since polynomials f and h are distinct it follow that $A = B \pm xx^T$ for some nonzero vector x. Now

$$f(z) = \det(zI - C_f) = \det(zI - A),$$

$$h(z) = \det(zI - C_h) = \det(zI - B),$$

and

$$\frac{h(z)}{f(z)} = \det\left((zI - B)(zI - A)^{-1}\right) = \det\left(I \pm xx^T(zI - A)^{-1}\right) = 1 \pm x^T(zI - A)^{-1}x.$$

This reduces, upon applying a diagonalizing real orthogonal transformation $A \mapsto U^T AU$ and replacing x with $U^T x$, to (2.1) with the real numbers c_j of the same sign, as required.

Finally, we prove the implication (3) $\implies$ (1). We have to show that if (3) holds true, then for any real γ the equation $\frac{h(z)}{f(z)} + \gamma = 0$ does not have roots with nonzero imaginary part. Consider a complex number $z = a + bi$, with the real part $\operatorname{Re}(z) = a$, and the imaginary part $\operatorname{Im}(z) = b$. If (3) holds, then

$$\operatorname{Im}\left(\frac{h(z)}{f(z)}\right) = \operatorname{Im}\left(\sum_{j=1}^{p} \frac{c_j}{z - \lambda_j}\right),$$

where the λ_j's are real and the real numbers c_j's are all of the same sign. As

$$\operatorname{Im}((z - \lambda_j)^{-1}) = \frac{-b}{(a - \lambda_j)^2 + b^2},$$

it follows that

$$\operatorname{Im}\left(\frac{h(z)}{f(z)}\right) = -b \sum_{j=1}^{p} \frac{c_j}{(a - \lambda_j)^2 + b^2}.$$

Therefore, $\operatorname{Im}(\frac{h(z)}{f(z)}) \neq 0$ if $\operatorname{Im}(z) \neq 0$. This means that the equation $\frac{h(z)}{f(z)} + \gamma = 0$ does not have roots with nonzero imaginary part for any real γ. ∎

We observe that checking condition (3) can be conveniently done using semidefinite programming. Indeed, let $h(z)$ and $f(z)$ be monic scalar polynomials with $f(z)$ having all roots real, and consider a minimal realization

$$\frac{h(z)}{f(z)} = 1 + \widetilde{C}(zI - \widetilde{A})^{-1}\widetilde{B},$$

where $\widetilde{C}$, $\widetilde{A}$, and $\widetilde{B}$ are real matrices. It is easy to see, using the uniqueness of a minimal realization up to a state isomorphism (similarity), that (3) holds, with the c_j's positive, if and only if there exists a positive definite matrix P such that

$$\widetilde{A}P = P\widetilde{A}^T, \qquad P\widetilde{C}^T = \widetilde{B}.$$

The latter problem is a semidefinite programming problem.

Another equivalent semidefinite programming problem is based on the following nice reformulation of Proposition 2.1: *The conditions of Proposition* 2.1 *are equivalent to the existence of nonsingular real matrix* D *such that both* DC_fD^{-1} *and* DC_hD^{-1} *are real symmetric.* (A proof of this statement is essentially the same rank one perturbation argument as in the proof of Proposition 2.1.) This amounts to the following semidefinite programming problem: *Is there a real positive definite matrix* P *such that the equalities*

$$PC_f = C_f^T P \quad \textit{and} \quad PC_h = C_h^T P$$

hold?

Our next corollary involves hyperbolic matrix polynomials.

COROLLARY 2.2. *Let* $L(z)$ *be a hyperbolic matrix polynomial. Then there exist* $n\ell \times n\ell$ *real symmetric matrices* A *and* B *such that the roots of* $\det(L(z) + tL'(z))$ *coincide with the eigenvalues of* $A + tB$ (*multiplicities counted*) *for every real number* t. *Here,* $L'(z)$ *is the derivative of* $L(z)$ *with respect to* z.

Proof. By Obreschkoff's theorem, the matrix polynomial $L(z) + tL'(z)$ is hyperbolic for every real t. Now apply Theorem 1.1 with $M(z) = L(z) + L'(z)$. ∎

Note that the condition (1.2) implies (but is not equivalent to) the condition that every convex combination of $L(z)$ and $M(z)$ is weakly hyperbolic. It turns out that the latter condition can be conveniently expressed for hyperbolic matrix polynomials, which we will do next.

Let $L(\lambda)$ be a hyperbolic $n \times n$ matrix polynomial. For every $x \in \mathbb{C}^n$, $\|x\| = 1$, let

$$\lambda_1(x) \le \lambda_2(x) \le \cdots \le \lambda_\ell(x)$$

be the roots of equation (1.1) arranged in nondecreasing order. The sets

$$\Delta_j(L) := \{\, \lambda_j(x) \,|\, x \in \mathbb{C}^n,\ \|x\| = 1\},$$

called the *spectral zones* of $L(\lambda)$, are obviously closed intervals on the real line:

$$\Delta_j(L) = [\delta_j^-(L), \delta_j^+(L)], \qquad j = 1, 2, \ldots, \ell.$$

A basic result in the theory of hyperbolic matrix and operator polynomials ([14, Theorem 31.5], for example), states that two spectral zones either are disjoint or have only one point in common.

Sufficient conditions for combinations of hyperbolic matrix polynomials being again hyperbolic can be given in terms of the spectral zones, as follows.

PROPOSITION 2.3. *Let* $L(\lambda)$ *and* $M(\lambda)$ *be two hyperbolic matrix polynomials of degree* ℓ.

(a) *Assume that the spectral zones of* L *and* M *satisfy the inequalities*

$$\max\{\delta_j^+(L), \delta_j^+(M)\} \le \min\{\delta_{j+1}^-(L), \delta_{j+1}^-(M)\}, \qquad j = 1, \ldots, \ell - 1.$$

Then every convex combination $\alpha L(z) + (1-\alpha)M(z)$, $0 \le \alpha \le 1$, *is hyperbolic.*

(b) *Assume that the spectral zones of* L *and* M *interlace, i.e., satisfy the inequalities*

$$\delta_j^+(L) \le \delta_j^-(M), \quad j = 1, \ldots, \ell, \quad \text{and} \quad \delta_j^+(M) \le \delta_{j+1}^-(L), \quad j = 1, \ldots, \ell - 1, \tag{2.2}$$

or the inequalities (2.2) *with the roles of* L *and* M *reversed. Then every combination* $\alpha L(z) + (1-\alpha)M(z)$, $\alpha \in \mathbb{R}$, *is hyperbolic.*

Proof. For the proof of (a) apply [3, Theorem 2.1]; this theorem gives necessary and sufficient conditions for all convex combinations of two given scalar polynomials to be hyperbolic. The proof of (b) follows from Obreschkoff's theorem and from the property that the spectral zones have at most one point in common. ∎

Using Theorem 1.1 and inequalities for eigenvalues of real symmetric matrices (see, for example, [12]), one can derive inequalities for eigenvalues of weakly hyperbolic matrix polynomials. We illustrate this for the case of the Horn inequalities. For a Hermitian $m \times m$ matrix X, we write its eigenvalues (repeated according to their multiplicities) in nondecreasing order:

$$\lambda_1(X) \le \lambda_2(X) \le \cdots \le \lambda_m(X).$$

An ordered triple (U, S, T) of nonempty subsets of $\{1, 2, \ldots, m\}$ is said to be a *Horn triple* (with respect to m) if the cardinalities of U, S, and T are the same, and the *Horn inequalities*

$$\sum_{i \in U} \lambda_i(X+Y) \le \sum_{j \in S} \lambda_j(X) + \sum_{k \in T} \lambda_k(Y)$$

hold true for every pair of Hermitian $m \times m$ matrices X and Y. A description of all Horn triples is known [8, 9]; see also the surveys [4, 2]. For a weakly hyperbolic $n \times n$ matrix polynomial $L(z)$ of degree ℓ, we arrange the roots of $\det(L(z))$ in nondecreasing order:

$$d_1(L) \le d_2(L) \le \cdots \le d_{n\ell}(L).$$

Let $T = \{i_1 < i_2 < \cdots < i_m\} \subseteq \{1, 2, \ldots, n\ell\}$; then we define

$$\overline{T} = \{n\ell + 1 - i_m < n\ell + 1 - i_{m-1} < \cdots < n\ell + 1 - i_1\}.$$

THEOREM 2.4. *Let* $L(z)$ *and* $M(z)$ *be monic* $n \times n$ *matrix polynomials satisfying the hypotheses of Theorem* 1.1. *Then for every Horn triple* (U, S, T) *with respect to* $n\ell$, *and for every* $\alpha \in \mathbb{R}$, *the inequality*

$$\sum_{i \in U} d_i\left(\alpha L + (1-\alpha)M\right) \le \alpha \left(\sum_{j \in S_\alpha} d_j(L) \right) + (1-\alpha) \left(\sum_{k \in T_{1-\alpha}} d_k(M) \right)$$

holds true. Here $S_\alpha = S$, $T_\alpha = T$ *if* $\alpha \ge 0$, *and* $S_\alpha = \overline{S}$, $T_\alpha = \overline{T}$ *if* $\alpha < 0$.

Proof. Let A and B be as in Theorem 1.1. Then we have, using Theorem 1.1 and the Horn inequalities,

$$\begin{aligned}
\sum_{i \in U} d_i\left(\alpha L + (1-\alpha)M\right) &= \sum_{i \in U} \lambda_i\left(\alpha A + (1-\alpha)B\right) \\
&\le \sum_{j \in S} \lambda_j\left(\alpha A\right) + \sum_{k \in T} \lambda_k\left((1-\alpha)B\right) \\
&= \alpha \left(\sum_{j \in S_\alpha} \lambda_j(A) \right) + (1-\alpha) \left(\sum_{k \in T_{1-\alpha}} \lambda_k(B) \right) \\
&= \alpha \left(\sum_{j \in S_\alpha} d_j(L) \right) + (1-\alpha) \left(\sum_{k \in T_{1-\alpha}} d_k(M) \right),
\end{aligned}$$

and the proof is complete. ∎

REFERENCES

[1] L. Barkwell, P. Lancaster, and A. S. Markus, Gyroscopically stabilized systems: A class of quadratic eigenvalue problems with real spectrum, Canad. J. Math., 44 (1992), pp. 42–53.
[2] R. Bhatia, *Linear algebra to quantum cohomology: The story of Alfred Horn's inequalities*, Amer. Math. Monthly, 108 (2001), pp. 289–318.
[3] J. P. Dedieu, *Obreschkoff's theorem revisited: What convex sets are contained in the set of hyperbolic polynomials?*, J. Pure Appl. Algebra, 81 (1992), pp. 269–278.
[4] W. Fulton, *Eigenvalues, invariant factors, highest weights, and Schubert calculus*, Bull. Amer. Math. Soc. (N.S.), 37 (2000), pp. 209–249.
[5] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Academic Press, New York, London, 1982.
[6] L. Gurvits, *Combinatorics Hidden in Hyperbolic Polynomials and Related Topics*, preprint, Los Alamos National Laboratory, Los Alamos, NM, 2004; also available from http://xxx.lanl.gov/abs/math.CO/0402088.
[7] J. W. Helton and V. Vinnikov, *Linear matrix inequality representation of sets*, preprint, University of California–San Diego, La Jolla, CA, 2003.
[8] A. A. Klyachko, *Stable bundles, representation theory, and Hermitian operators*, Selecta Math. (N.S.), 3 (1998), pp. 419–445.
[9] A. A. Klaychko, *Vector bundles, linear representations, and spectral problems*, in Proceedings of the International Congress of Mathematicians, Vol. II, Higher Ed. Press, Beijing, 2002, pp. 599–613.
[10] P. Lancaster, A. S. Markus, and V. I. Matsaev, *Definitizable operators and quasihyperbolic operator polynomials*, J. Funct. Anal., 131 (1995), pp. 1–28.
[11] P. D. Lax, *Differential equations, difference equations, and matrix theory*, Comm. Pure Appl. Math., 11 (1958), pp. 175–194.
[12] A. S. Lewis and M. L. Overton, *Eigenvalue optimization*, in Acta Numerica, Acta Numer. 5, Cambridge University Press, Cambridge, UK, 1996, pp. 149–190.
[13] A. S. Lewis, P. A. Parrilo, and M. V. Ramana, *The Lax conjecture is true*, Proc. Amer. Math. Soc., to appear.
[14] A. S. Markus, *Introduction to the spectral theory of polynomial operator pencils*, Trans. Math. Monogr., 71, AMS, Providence, RI, 1988.
[15] A. S. Markus, V. I. Macaev, and G. I. Russu, *Certain generalizations of the theory of strongly damped pencils to the case of pencils of arbitrary order*, Acta Sci. Math. (Szeged), 34 (1973), pp. 245–271.
[16] N. Obreschkoff, *Verteilung und Berechnung der Nullstellen reeler Polynome*, VEB Deutscher Verlag der Wissenschaften, Berlin, 1963.
[17] V. Vinnikov, *Selfadjoint determinantal representations of real plane curves*, Math. Ann., 296 (1993), pp. 453–479.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 3, pp. 765–781

QUADRATURE RULES BASED ON THE ARNOLDI PROCESS*

DANIELA CALVETTI†, SUN-MI KIM‡, AND LOTHAR REICHEL‡

Abstract. Applying a few steps of the Arnoldi process to a large nonsymmetric matrix A with initial vector v is shown to induce several quadrature rules. Properties of these rules are discussed, and their application to the computation of inexpensive estimates of the quadratic form $\langle f, g\rangle := v^*(f(A))^* g(A)v$ and related quadratic and bilinear forms is considered. Under suitable conditions on the functions f and g, the matrix A, and the vector v, the computed estimates provide upper and lower bounds of the quadratic and bilinear forms.

Key words. quadrature method, bilinear form, error estimate, large-scale problem

AMS subject classifications. 65D32, 65F10

DOI. 10.1137/S0895479803423822

1. Introduction.

Let $A \in \mathbb{C}^{N\times N}$ be a large, possibly sparse, nonsymmetric matrix and let the vector $v \in \mathbb{C}^N$ be nonvanishing. Application of $n \ll N$ steps of the Arnoldi process to the matrix A with initial vector v yields the Arnoldi decomposition

$$AV_n = V_n H_n + \hat{v}_{n+1} e_n^*, \tag{1.1}$$

where $V_n = [v_1, v_2, \ldots, v_n] \in \mathbb{C}^{N\times n}$ and $\hat{v}_{n+1} \in \mathbb{C}^N$ satisfy $V_n^* V_n = I_n$, $V_n^* \hat{v}_{n+1} = 0$, $v_1 = v/\|v\|$, and $H_n = [h_{ij}]_{i,j=1}^n \in \mathbb{C}^{n\times n}$ is an upper Hessenberg matrix with nonvanishing subdiagonal entries $h_{i+1,i}$, $1 \le i < n$. Here and throughout this paper I_n denotes the $n \times n$ identity matrix, e_j denotes the jth axis vector of appropriate dimension, $\|\cdot\|$ denotes the Euclidean vector norm, and the superscript $*$ denotes transposition and, if applicable, complex conjugation. We tacitly assume that the number of steps n of the Arnoldi process is small enough so that the decomposition (1.1) with the stated properties exists; see, e.g., Golub and Van Loan [11, Chapter 9] or Saad [16, Chapter 6] for discussions on the Arnoldi process. Here we note only that the evaluation of the Arnoldi decomposition (1.1) requires the computation of n matrix-vector products with the matrix A. The fact that A does not have to be factored makes it possible to compute Arnoldi decompositions (1.1) for small to moderate values of n also when the order N of the matrix A is very large.

We are particularly interested in the generic case when $\hat{v}_{n+1}$ is nonvanishing; when $\hat{v}_{n+1} = 0$ our discussion simplifies. Thus, unless explicitly stated otherwise, we assume that $\hat{v}_{n+1} \neq 0$ and define

$$h_{n+1,n} := \|\hat{v}_{n+1}\|, \qquad v_{n+1} := \hat{v}_{n+1}/h_{n+1,n}. \tag{1.2}$$

Introduce the quadratic form

$$\langle f, g\rangle := v^*(f(A))^* g(A)v \tag{1.3}$$

*Received by the editors March 2, 2003; accepted for publication (in revised form) by D. Boley August 2, 2004; published electronically April 8, 2005.
http://www.siam.org/journals/simax/26-3/42382.html

†Department of Mathematics, Case Western Reserve University, Cleveland, OH 44106 (daniela.calvetti@case.edu). This research was supported in part by NSF grant DMS-0107841.

‡Department of Mathematical Sciences, Kent State University, Kent, OH 44242 (sukim@math.kent.edu, reichel@math.kent.edu). This research was supported in part by NSF grant DMS-0107858.

for functions f and g that are analytic in a neighborhood of the eigenvalues of A. The analyticity requirements on f and g are satisfied in many applications and allow us to use the Jordan normal form of A to define $f(A)$ and $g(A)$; see, e.g., Gantmacher [8, Chapter 5] for details. We remark that the quadratic form (1.3) is positive semidefinite.

The quadratic form (1.3) can also be represented as the double integral

$$\langle f, g\rangle = \frac{1}{4\pi^2}\int_\Gamma\int_\Gamma \overline{f(z_1)}g(z_2)v^*(\overline{z}_1 I - A^*)^{-1}(z_2 I - A)^{-1}v\overline{dz}_1 dz_2, \tag{1.4}$$

where the contour of integration Γ contains the spectrum of A in its interior and the bar denotes complex conjugation; see, e.g., [12] for discussions of related representations.

The present paper is concerned with the approximation of the quadratic form (1.3) and related quadratic and bilinear forms by expressions that are based on the Arnoldi decomposition (1.1) and are easy to evaluate when $n \ll N$. For instance, we consider the approximation of (1.3) by the positive semidefinite quadratic form

$$\langle f, g\rangle_n := \|v\|^2 e_1^*(f(H_n))^* g(H_n)e_1, \tag{1.5}$$

where the functions f and g also are required to be analytic in a neighborhood of the eigenvalues of H_n. The expression (1.5) can be considered a quadrature rule for approximating the integral (1.4), and we refer to (1.5) as an *Arnoldi quadrature rule*. The error $\langle f, g\rangle - \langle f, g\rangle_n$ has been investigated by Freund and Hochbruck [7]. We review their results in section 2.

A new quadrature rule,

$$[f, g]_{n+1} := \|v\|^2 e_1^*(f(\tilde{H}_{n+1}))^* g(\tilde{H}_{n+1})e_1, \tag{1.6}$$

for the approximation of (1.3) is introduced in section 3. The matrix $\tilde{H}_{n+1}$ in (1.6) is defined as follows. Let H_{n+1} be the upper Hessenberg matrix in (1.1) with n replaced by $n+1$. Since the entry $h_{n+1,n}$ is assumed to be nonvanishing, cf. (1.2), the matrix H_{n+1} exists. The matrix $\tilde{H}_{n+1} \in \mathbb{C}^{(n+1)\times(n+1)}$ in (1.6) now is determined by modifying some of the entries in H_{n+1}, so that

$$\langle f, g\rangle - \langle f, g\rangle_n = -(\langle f, g\rangle - [f, g]_{n+1}) \qquad \forall\{f, g\} \in \mathbb{W}_n, \tag{1.7}$$

where

$$\mathbb{W}_n := (\mathbb{P}_n \oplus \mathbb{P}_{n+1}) \cup (\mathbb{P}_{n+1} \oplus \mathbb{P}_n) \tag{1.8}$$

and $\mathbb{P}_j$ denotes the set of all polynomials of degree at most j. Because of the property (1.7), we refer to (1.6) as an *anti-Arnoldi quadrature rule*. This rule generalizes the anti-Gauss rules introduced by Laurie [14]. Application of the latter rules to the estimation of functionals of the form (1.3) with a Hermitian matrix A is discussed in [5, 6].

Section 4 considers expansions of f and g in terms of certain orthogonal polynomials determined by the Arnoldi process. We show that if these expansions converge sufficiently rapidly, then the real and imaginary parts of $\langle f, g\rangle_n$ and $[f, g]_{n+1}$ furnish upper and lower bounds, or lower and upper bounds, respectively, of the real and imaginary parts of $\langle f, g\rangle$. This is illustrated by computed examples in section 5.

When $f(t) := 1$, the quadratic form (1.3) simplifies to the functional

$$\mathcal{I}(g) := v^* g(A) v. \tag{1.9}$$

The approximation of functionals of the form (1.9), when the matrix A is Hermitian, has received considerable attention; see, e.g., [1, 2, 3, 4, 10, 15]. These references exploit the connection between the Hermitian Lanczos process, orthogonal polynomials, and Gauss quadrature rules. A nice survey of these techniques is provided by Golub and Meurant [9]. In the present paper, we are concerned with the approximation of functionals of the form (1.9) with a non-Hermitian matrix A. Application of the non-Hermitian Lanczos process to this problem, using the connection with biorthogonal polynomials, is described in [6, 17, 18]. Knizhnerman [13] considers application of the Arnoldi process to the approximation of $f(A)v$ and discusses the rate of convergence.

We conclude this section with a few applications, where the computation of inexpensive estimates of quantities of the form (1.3) or (1.9) is desirable. The problem of computing approximations of expressions of the form $u^* A^{-1} v$, where A is a large matrix and u and v are given vectors, arises in electromagnetic scattering; see Saylor and Smolarski [17, 18] for a discussion. The technique of the present paper is attractive to use when an Arnoldi decomposition (1.1) has been determined and $u = f(A)v$ for an explicitly known function f, which is analytic in a neighborhood of the spectrum of A. We apply the quadrature rules (1.5) and (1.6) with f and $g(t) := 1/t$ to determine the desired estimates. Estimates also can be computed without using the function f if u can be expressed as a linear combination of the first few columns of the matrix V_n in (1.1). This is discussed in section 4.

Estimates of $v_j^* A^{-1} v_j$, $1 \le j \le k$, where v_j denotes the jth column of the matrix V_n, can be used for constructing preconditioners for linear systems of equations $Ax = v$. Let d_j be computed estimates of $v_j^* A^{-1} v_j$. Then

$$\begin{aligned} M &= V_k \mathrm{diag}[d_1, d_2, \ldots, d_k] V_k^* + (I - V_k V_k^*) \\ &= I + V_k \mathrm{diag}[d_1 - 1, d_2 - 1, \ldots, d_k - 1] V_k^* \end{aligned}$$

can be used as a preconditioner. Estimates d_j can be computed inexpensively, e.g., by using (1.5), (1.6), or the average of the two, with $f(t) := 1$ and $g(t) := 1/t$, if the linear system of equations is solved by the restarted GMRES method and therefore the Arnoldi decomposition (1.1) is available. Knowledge of estimates of upper and lower bounds of the quantities $v_j^* A^{-1} v_j$ makes it possible to assess the accuracy of the d_j.

Finally, consider the linear continuous-time system

$$\begin{aligned} x'(t) &= Ax(t) + vs(t), \\ y(t) &= u^* x(t), \end{aligned}$$

where $u, v \in \mathbb{R}^N$, $'$ denotes differentiation with respect to time t, $x(0) := 0$, and s is a real-valued function. The impulse response of this system (with $s(t)$ the Dirac δ-function) is given by

$$h(t) := u^* \exp(At) v, \qquad t \ge 0,$$

see, e.g., [19, section 2.7]. The techniques developed in this paper can be used to compute estimates of upper and lower bounds of h under the conditions on u stated above.

2. Arnoldi quadrature rules. Identification of the columns in the right-hand side and left-hand side of (1.1) and using (1.2) show that

$$v_j = \hat{p}_{j-1}(A)v, \qquad 1 \le j \le n+1, \tag{2.1}$$

for certain uniquely determined polynomials $\hat{p}_{j-1}$. Here $\hat{p}_{j-1}$ is of degree $j-1$ and has a positive leading coefficient. Combining (2.1) and (1.3) yields

$$v_j^* v_k = \langle \hat{p}_{j-1}, \hat{p}_{k-1} \rangle, \qquad 1 \le j,k \le n+1,$$

and it follows from the orthonormality of the vectors v_j generated by the Arnoldi process that the polynomials $\hat{p}_{j-1}$ are orthonormal with respect to the quadratic form (1.3). Substituting (2.1) into (1.1) and (1.2) shows that the polynomials $\hat{p}_{j-1}$, $1 \le j \le n+1$, satisfy a recursion relation, whose coefficients are determined by the entries of the matrix H_n and by $h_{n+1,n}$.

It is convenient to work with the monic orthogonal polynomials p_{j-1} associated with the polynomials $\hat{p}_{j-1}$. Let $n(A)$ denote the grade of A with respect to v; i.e., $n(A)$ is the smallest positive integer with the property that there is a nonvanishing polynomial $p \in \mathbb{P}_{n(A)}$ such that $p(A)v = 0$. Note that the quadratic form (1.3) is an inner product on the space $\mathbb{P}_{n(A)-1} \oplus \mathbb{P}_{n(A)-1}$ because it is positive definite there.

PROPOSITION 2.1. *There is a family $\{p_j\}_{j=0}^{n(A)}$ of monic polynomials that are orthogonal with respect to the quadratic form* (1.3). *The polynomials satisfy the recurrence relation*

$$\begin{cases} p_0(t) = 1, \\ p_j(t) = (t - c_{jj})p_{j-1}(t) - \displaystyle\sum_{k=1}^{j-1} c_{kj} p_{k-1}(t), & 1 \le j \le n(A), \end{cases} \tag{2.2}$$

where

$$c_{kj} := \frac{\langle p_{k-1}, tp_{j-1} \rangle}{\langle p_{k-1}, p_{k-1} \rangle}, \qquad 1 \le k \le j \le n(A). \tag{2.3}$$

Moreover, for $1 \le n \le n(A)$, the nontrivial entries of the upper Hessenberg matrices H_n, and the scalar $h_{n+1,n}$ defined by (1.2), *can be expressed as*

$$\begin{cases} h_{j+1,j} = \dfrac{\langle p_j, p_j \rangle^{1/2}}{\langle p_{j-1}, p_{j-1} \rangle^{1/2}}, & 1 \le j \le n, \\ h_{kj} = \dfrac{\langle p_{k-1}, tp_{j-1} \rangle}{\langle p_{k-1}, p_{k-1} \rangle^{1/2} \langle p_{j-1}, p_{j-1} \rangle^{1/2}}, & 1 \le k \le j \le n. \end{cases} \tag{2.4}$$

In particular,

$$h_{j+1,j} > 0, \qquad 1 \le j < n(A), \tag{2.5}$$

and

$$h_{n(A)+1,n(A)} = 0. \tag{2.6}$$

Proof. Let the polynomials p_j be defined by (2.2) for some recursion coefficients c_{kj}. Then p_j is of degree j, and it follows from the definition of $n(A)$ that

$$\langle p_j, p_j \rangle > 0, \qquad 0 \le j < n(A). \tag{2.7}$$

Hence, the expressions (2.3) and (2.4) for the coefficients c_{kj}, h_{kj}, and $h_{j+1,j}$ are well defined for $1 \leq k \leq j \leq n(A)$. In particular, the family of polynomials $\{p_j\}_{j=0}^{n(A)}$ determined by the recursion coefficients (2.3) is well defined.

The fact that the polynomial p_j can be obtained by scaling the polynomials $\hat{p}_j$, defined by (2.1), and the relation (2.4) between the p_j and the entries h_{kj} of H_n and the scalar $h_{n+1,n}$ defined by (1.2) follows from the Arnoldi decomposition (1.1) and straightforward computations.

The equations (2.4) and inequalities (2.7) show (2.5). We turn to the proof of (2.6). Let the polynomial p of degree $n(A) > 0$ satisfy $p(A)v = 0$ and express p in the form

$$p(t) = \sum_{j=0}^{n(A)} \alpha_j p_j(t).$$

Then

$$0 = \langle p, p_k \rangle = \sum_{j=0}^{n(A)} \bar{\alpha}_j \langle p_j, p_k \rangle = \bar{\alpha}_k \langle p_k, p_k \rangle, \qquad 0 \leq k \leq n(A). \tag{2.8}$$

The inequalities (2.7) imply that $\alpha_k = 0$ for $0 \leq k < n(A)$, and therefore $p(t) = \alpha_{n(A)} p_{n(A)}(t)$. Since p is of exactly degree $n(A)$, the coefficient $\alpha_{n(A)}$ is nonvanishing and it follows from (2.8) that $\langle p_{n(A)}, p_{n(A)} \rangle = 0$. This shows (2.6). ∎

The following result has been shown by Freund and Hochbruck [7]. A proof is included for completeness.

THEOREM 2.2. *Let $n \leq n(A)$. Then*

$$\langle f, g \rangle_n = \langle f, g \rangle \qquad \forall \{f, g\} \in \mathbb{W}_{n-1}, \tag{2.9}$$

where the set $\mathbb{W}_{n-1}$ is defined by (1.8).

Proof. Since $\langle f, g \rangle = \overline{\langle g, f \rangle}$ and $\langle f, g \rangle_n = \overline{\langle g, f \rangle}_n$, where the bar denotes complex conjugation, it suffices to show that

$$\langle f, g \rangle_n = \langle f, g \rangle \qquad \forall f \in \mathbb{P}_n, \quad \forall g \in \mathbb{P}_{n-1}. \tag{2.10}$$

Clearly, this equality only has to be established for monomials $f(t) = t^k$ and $g(t) = t^j$. Thus, (2.10) is equivalent to

$$(H_n^k e_1)^* (H_n^j e_1) = (A^k v_1)^* (A^j v_1), \qquad 0 \leq k \leq n, \quad 0 \leq j \leq n-1. \tag{2.11}$$

The Arnoldi decomposition (1.1) and induction over j yield

$$\begin{cases} A^j v_1 = V_n H_n^j e_1, & j = 0, 1, 2, \ldots, n-1, \\ A^n v_1 = V_n H_n^n e_1 + \hat{v}_{n+1} e_n^* H_n^{n-1} e_1. \end{cases} \tag{2.12}$$

The relation (2.11) now follows from (2.12), $V_n^* V_n = I_n$ and $V_n^* \hat{v}_{n+1} = 0$. ∎

We next show that if the vector $\hat{v}_{n+1}$ in the Arnoldi decomposition (1.1) vanishes, then the Arnoldi quadrature rule (1.5) is exact for all polynomials.

COROLLARY 2.3. *Assume that $n = n(A)$. Then*

$$\langle f, g \rangle_n = \langle f, g \rangle$$

for all polynomials f and g.

Proof. It follows from (2.6) that the vector $\hat{v}_{n+1}$ in (1.1) vanishes. Therefore,

$$A^j v_1 = V_n H_n^j e_1$$

for all nonnegative integers j. This implies the desired result. ∎

It is convenient to introduce the bilinear forms

$$\langle f, g\rangle^{(r,s)} := \|v\|^2 v_r^*(f(A))^* g(A) v_s, \qquad 1 \le r, s \le n,$$

and

$$\langle f, g\rangle_n^{(r,s)} := \|v\|^2 e_r^*(f(H_n))^* g(H_n) e_s, \qquad 1 \le r, s \le n, \tag{2.13}$$

where v is the initial vector for the Arnoldi process and v_j is the jth column of the matrix V_n, i.e., $v_j = V_n e_j$ for $1 \le j \le n$; cf. (1.1). We note that $\langle f, g\rangle = \langle f, g\rangle^{(1,1)}$ and $\langle f, g\rangle_n = \langle f, g\rangle_n^{(1,1)}$.

THEOREM 2.4. *Let $n \le n(A)$. Then*

$$\langle f, g\rangle^{(r,s)} = \langle f, g\rangle_n^{(r,s)}$$

for all integers r, s such that $1 \le r, s \le n$, and all polynomials f, g such that either $f \in \mathbb{P}_{n-r+1}$ and $g \in \mathbb{P}_{n-s}$ or $f \in \mathbb{P}_{n-r}$ and $g \in \mathbb{P}_{n-s+1}$.

Proof. Let $\hat{f} \in \mathbb{P}_n$ and $\hat{g} \in \mathbb{P}_{n-1}$ be of the form

$$\hat{f} = \|v\| f \hat{p}_{r-1}, \qquad \hat{g} = \|v\| g \hat{p}_{s-1},$$

where the polynomials $\hat{p}_{r-1}$ and $\hat{p}_{s-1}$ are defined by (2.1). It follows from (2.1) that

$$\hat{f}(A)v = \|v\| f(A) v_r, \qquad \hat{g}(A)v = \|v\| g(A) v_s,$$

and therefore

$$\langle \hat{f}, \hat{g}\rangle = \langle f, g\rangle^{(r,s)}. \tag{2.14}$$

The proof of Theorem 2.2, specifically (2.12), shows that

$$V_n^* p(A) v_1 = p(H_n) e_1 \qquad \forall p \in \mathbb{P}_n. \tag{2.15}$$

Substituting the polynomials $\hat{p}_{j-1}$ defined by (2.1) into (2.15) yields

$$e_j = \|v\| \hat{p}_{j-1}(H_n) e_1, \qquad 1 \le j \le n, \tag{2.16}$$

and it follows that

$$\langle \hat{f}, \hat{g}\rangle_n = \langle f, g\rangle_n^{(r,s)}. \tag{2.17}$$

Combining (2.9), (2.14), and (2.17) shows that

$$\langle f, g\rangle^{(r,s)} = \langle f, g\rangle_n^{(r,s)}. \tag{2.18}$$

Similarly, when $\hat{f} \in \mathbb{P}_{n-1}$ and $\hat{g} \in \mathbb{P}_n$, (2.18) follows for $f \in \mathbb{P}_{n-r}$ and $g \in \mathbb{P}_{n-s+1}$. This establishes the theorem. ∎

COROLLARY 2.5. *Assume that $n = n(A)$. Then*

$$\langle f, g\rangle^{(r,s)} = \langle f, g\rangle_n^{(r,s)}$$

for all integers r and s such that $1 \le r, s \le n$ and for all polynomials f and g.

Proof. It follows from (2.6) that the vector $\hat{v}_{n+1}$ in (1.1) vanishes. Therefore,

$$A^k v_r = V_n H_n^k e_r$$

for all nonnegative integers k and $1 \le r \le n$. The desired result follows. ∎

3. Anti-Arnoldi quadrature rules. This section discusses the construction of the matrix $\tilde{H}_{n+1}$ in the anti-Arnoldi rule (1.6), which is characterized by (1.7). Our derivation of $\tilde{H}_{n+1}$ is analogous to the derivation of the symmetric tridiagonal matrices associated with anti-Gauss quadrature rules introduced by Laurie [14].

Equation (1.7) is equivalent to

$$[f,g]_{n+1} = 2\langle f,g\rangle - \langle f,g\rangle_n \qquad \forall\{f,g\} \in \mathbb{W}_n. \tag{3.1}$$

Hence, the quadratic form $[f,g]_{n+1}$ defined by (1.6) can be considered an Arnoldi quadrature rule associated with the quadratic form

$$[f,g] := 2\langle f,g\rangle - \langle f,g\rangle_n \tag{3.2}$$

defined for all functions f and g that are analytic in a neighborhood of all eigenvalues of A and H_n. A comparison of (3.1) and (3.2) shows that the quadratic form defined by (1.6) satisfies

$$[f,g]_{n+1} = [f,g] \qquad \forall\{f,g\} \in \mathbb{W}_n. \tag{3.3}$$

We are particularly interested in the case when $n < n(A)$, because $n = n(A)$ implies that the vector $\hat{v}_{n+1}$ in (1.1) vanishes, and then, by Corollary 2.3, $[f,g] = \langle f,g\rangle$ for all polynomials f and g.

PROPOSITION 3.1. *Let $n \le n(A)$. Then there is a family $\{\tilde{p}_j\}_{j=0}^{n}$ of monic polynomials that are orthogonal with respect to the quadratic form* (3.2). *The polynomials satisfy*

$$\begin{cases} \tilde{p}_0(t) = 1, \\ \tilde{p}_j(t) = (t - \tilde{c}_{jj})\tilde{p}_{j-1}(t) - \displaystyle\sum_{k=1}^{j-1} \tilde{c}_{kj}\tilde{p}_{k-1}(t), \qquad 1 \le j \le n, \end{cases} \tag{3.4}$$

where

$$\tilde{c}_{kj} := \frac{[\tilde{p}_{k-1}, t\tilde{p}_{j-1}]}{[\tilde{p}_{k-1}, \tilde{p}_{k-1}]}, \qquad 1 \le k \le j \le n. \tag{3.5}$$

Proof. Formulas (3.4) and (3.5) are analogous to (2.2) and (2.3), respectively. The latter formulas express orthogonality with respect to the quadratic form (1.3), and similarly the formulas (3.4) and (3.5) express orthogonality with respect to the quadratic form (3.2), provided that the denominators in (3.5) do not vanish for $1 \le k \le n$. We now establish the latter. It follows from (3.2), Theorem 2.2, and $n \le n(A)$ that

$$[f,f] = \langle f,f\rangle > 0 \qquad \forall f \in \mathbb{P}_{n-1}.$$

This shows the proposition. ∎

We are now in a position to define the upper Hessenberg matrix

$$\tilde{H}_{n+1} = [\tilde{h}_{jk}]_{j,k=1}^{n+1} \in \mathbb{C}^{(n+1)\times(n+1)}$$

in (1.6) using formulas analogous to (2.4). Thus, let

$$\tilde{h}_{j+1,j} := \frac{[\tilde{p}_j, \tilde{p}_j]^{1/2}}{[\tilde{p}_{j-1}, \tilde{p}_{j-1}]^{1/2}}, \qquad 1 \le j \le n, \tag{3.6}$$

$$\tilde{h}_{kj} := \frac{[\tilde{p}_{k-1}, t\tilde{p}_{j-1}]}{[\tilde{p}_{k-1}, \tilde{p}_{k-1}]^{1/2}[\tilde{p}_{j-1}, \tilde{p}_{j-1}]^{1/2}}, \qquad 1 \le k \le j \le n+1, \tag{3.7}$$

$$\tilde{h}_{kj} := 0, \qquad 2 \le j+1 < k \le n+1.$$

Let $H_{n+1} = [h_{kj}]_{k,j=1}^{n+1}$ be the upper Hessenberg matrix of order $n+1$ determined by $n+1$ steps of the Arnoldi process applied to A with initial vector v. The matrix H_n is the leading principal submatrix of H_{n+1} of order n. The following theorem shows that

$$\tilde{H}_{n+1} = \begin{bmatrix} & & & \sqrt{2}h_{1,n+1} \\ & H_n & & \sqrt{2}h_{2,n+1} \\ & & & \vdots \\ & & & \sqrt{2}h_{n,n+1} \\ 0 & \cdots & 0 & \sqrt{2}h_{n+1,n} \quad h_{n+1,n+1} \end{bmatrix} \in \mathbb{C}^{(n+1)\times(n+1)}.$$

THEOREM 3.2. *Assume that $n < n(A)$ and let h_{kj}, $1 \le k, j \le n+1$, be the entries of the upper Hessenberg matrix H_{n+1} in the Arnoldi decomposition* (1.1) *with n replaced by $n+1$. Then*

$$\tilde{h}_{kj} = h_{kj}, \qquad 1 \le k, j \le n, \tag{3.8}$$

$$\begin{aligned} \tilde{h}_{j+1,j} &= h_{j+1,j}, \qquad 1 \le j < n, \\ \tilde{h}_{k,n+1} &= \sqrt{2}h_{k,n+1}, \qquad 1 \le k \le n, \\ \tilde{h}_{n+1,n} &= \sqrt{2}h_{n+1,n}, \\ \tilde{h}_{n+1,n+1} &= h_{n+1,n+1}. \end{aligned} \tag{3.9}$$

Proof. It follows from Proposition 2.1 that the entries h_{kj} of the matrix H_{n+1} are well defined. Theorem 2.2 and (3.2) yield

$$[f, g] = \langle f, g \rangle \qquad \forall \{f, g\} \in \mathbb{W}_{n-1}, \tag{3.10}$$

and therefore it follows from the recurrence relations (2.2)–(2.3) and (3.4)–(3.5) that

$$\tilde{c}_{kj} = c_{kj}, \qquad 1 \le k \le j \le n,$$

and

$$\tilde{p}_j = p_j, \qquad 0 \le j \le n. \tag{3.11}$$

Formulas (3.8) and (3.9) follow from relations (2.4), (3.6), and (3.7), and equalities (3.10) and (3.11).

The polynomial p_n is the characteristic polynomial of H_n. This can be seen as follows. Equation (2.15) shows that

$$p_n(H_n)e_1 = 0. \tag{3.12}$$

This equation and the fact that H_n has positive subdiagonal entries determine the coefficients in the representation

$$p_n(t) = t^n + \sum_{j=0}^{n-1} \alpha_j t^j$$

uniquely. By the Cayley–Hamilton theorem, the characteristic polynomial of H_n satisfies (3.12). Hence, p_n is the characteristic polynomial and it follows that

$$\langle p, qp_n \rangle_n = 0$$

for any polynomials p and q. Therefore, from (2.4), (3.2), (3.6), (3.7), (3.10), and (3.11), we obtain

$$\tilde{h}_{k,n+1} = \frac{2\langle p_{k-1}, tp_n\rangle}{\sqrt{2}\langle p_{k-1}, p_{k-1}\rangle^{1/2}\langle p_n, p_n\rangle^{1/2}} = \sqrt{2}h_{k,n+1}, \qquad 1 \le k \le n,$$
$$\tilde{h}_{n+1,n} = \frac{\sqrt{2}\langle p_n, p_n\rangle^{1/2}}{\langle p_{n-1}, p_{n-1}\rangle^{1/2}} = \sqrt{2}h_{n+1,n},$$
$$\tilde{h}_{n+1,n+1} = \frac{2\langle p_n, tp_n\rangle}{2\langle p_n, p_n\rangle} = h_{n+1,n+1}.$$

This shows the theorem. □

Introduce the bilinear forms

$$[f,g]^{(r,s)} := 2\langle f,g\rangle^{(r,s)} - \langle f,g\rangle_n^{(r,s)}, \qquad 1 \le r,s \le n, \tag{3.13}$$

and

$$[f,g]_{n+1}^{(r,s)} := \|v\|^2 e_r^*(f(\tilde{H}_{n+1}))^* g(\tilde{H}_{n+1})e_s, \qquad 1 \le r,s \le n, \tag{3.14}$$

where we note that $[f,g]^{(1,1)} = [f,g]$ and $[f,g]_{n+1}^{(1,1)} = [f,g]_{n+1}$. The following result is analogous to Theorem 2.4.

THEOREM 3.3. *Assume that $n < n(A)$. Then*

$$[f,g]_{n+1}^{(r,s)} = [f,g]^{(r,s)}$$

for all integers r, s such that $1 \le r,s \le n$, and for all polynomials f, g such that either $f \in \mathbb{P}_{n-r+2}$ and $g \in \mathbb{P}_{n-s+1}$ or $f \in \mathbb{P}_{n-r+1}$ and $g \in \mathbb{P}_{n-s+2}$.

Proof. Let $\hat{f} \in \mathbb{P}_{n+1}$ and $\hat{g} \in \mathbb{P}_n$ be of the form

$$\hat{f} = \|v\| f\hat{p}_{r-1}, \qquad \hat{g} = \|v\| g\hat{p}_{s-1},$$

where the $\hat{p}_{r-1}$ and $\hat{p}_{s-1}$ are defined by (2.1). Similar to the proof of Theorem 2.4, we obtain that $\langle \hat{f},\hat{g}\rangle = \langle f,g\rangle^{(r,s)}$ and $\langle f,g\rangle_n = \langle f,g\rangle_n^{(r,s)}$ for $1 \le r,s \le n$. Therefore, by (3.2) and (3.13),

$$[\hat{f},\hat{g}] = 2\langle \hat{f},\hat{g}\rangle - \langle \hat{f},\hat{g}\rangle_n = 2\langle f,g\rangle^{(r,s)} - \langle f,g\rangle_n^{(r,s)} = [f,g]^{(r,s)}. \tag{3.15}$$

Application of $n+1$ steps of the Arnoldi process to the matrix A with initial vector v yields an Arnoldi decomposition analogous to (1.1) with n replaced by $n+1$. Let $H_{n+1} \in \mathbb{C}^{(n+1)\times(n+1)}$ denote the upper Hessenberg matrix in this decomposition. Then, analogously to (2.16), we have

$$e_j = \|v\|\hat{p}_{j-1}(H_{n+1})e_1, \qquad 1 \le j \le n.$$

Straightforward computation yields

$$\tilde{H}_{n+1}^k e_1 = H_{n+1}^k e_1, \qquad 0 \le k \le n-1,$$

and therefore

$$\hat{p}_{j-1}(\tilde{H}_{n+1})e_1 = \hat{p}_{j-1}(H_{n+1})e_1, \qquad 1 \le j \le n.$$

Hence, $\|v\|\hat{p}_{j-1}(\tilde{H}_{n+1})e_1 = e_j$, $1 \le j \le n$, and it follows that for $1 \le r, s \le n$,

$$\begin{aligned}[\hat{f},\hat{g}]_{n+1} &= \|v\|^2 e_1^*(\hat{f}(\tilde{H}_{n+1}))^*\hat{g}(\tilde{H}_{n+1})e_1 \\ &= \|v\|^2 e_r^*(f(\tilde{H}_{n+1}))^* g(\tilde{H}_{n+1})e_s = [f,g]_{n+1}^{(r,s)}.\end{aligned} \tag{3.16}$$

Combining (3.3), (3.15), and (3.16) shows that

$$[f,g]_{n+1}^{(r,s)} = [f,g]^{(r,s)}, \qquad 1 \le r, s \le n. \tag{3.17}$$

When instead $\hat{f} \in \mathbb{P}_n$ and $\hat{g} \in \mathbb{P}_{n+1}$, (3.17) can be shown in a similar fashion for $f \in \mathbb{P}_{n-r+1}$ and $g \in \mathbb{P}_{n-s+2}$. This completes the proof of the theorem. ∎

4. Applications of Arnoldi and anti-Arnoldi quadrature rules. Let f and g be functions such that $f(A)v$ and $g(A)v$ have the expansions

$$f(A)v = \sum_{i=0}^{m_1} \eta_i \hat{p}_i(A)v, \qquad g(A)v = \sum_{j=0}^{m_2} \xi_j \hat{p}_j(A)v, \tag{4.1}$$

where m_1 and m_2 are nonnegative integers such that $\max\{m_1, m_2\} \le n(A)$. For notational simplicity, we assume that n, the number of steps of the Arnoldi process, is at most $\min\{m_1, m_2\} - 1$. Then

$$\langle f, g\rangle = \sum_{i=0}^{m_1}\sum_{j=0}^{m_2} \bar{\eta}_i \xi_j \langle \hat{p}_i, \hat{p}_j\rangle, \tag{4.2}$$

and the orthonormality of the polynomials $\hat{p}_i$ yields

$$\langle f, g\rangle = \sum_{i=0}^{n-1} \bar{\eta}_i \xi_i + \bar{\eta}_n \xi_n + \sum_{i=n+1}^{\min\{m_1,m_2\}} \bar{\eta}_i \xi_i. \tag{4.3}$$

Replacing the quadratic form $\langle\cdot,\cdot\rangle$ by $\langle\cdot,\cdot\rangle_n$ in (4.2), and using the facts that $\langle \hat{p}_i, \hat{p}_j\rangle_n = \langle \hat{p}_i, \hat{p}_j\rangle$ for $0 \le i, j < n$, and $\langle \hat{p}_n, q\rangle_n = 0$ for any polynomial q, shows that

$$\begin{aligned}\langle f, g\rangle_n = &\sum_{i=0}^{n-1} \bar{\eta}_i \xi_i + \sum_{i=0}^{n-1} (\bar{\eta}_i \xi_{n+1} \langle \hat{p}_i, \hat{p}_{n+1}\rangle_n + \bar{\eta}_{n+1}\xi_i \langle \hat{p}_{n+1}, \hat{p}_i\rangle_n) \\ &+ \sum_{j=n+1}^{m_2} \bar{\eta}_{n+1}\xi_j \langle \hat{p}_{n+1}, \hat{p}_j\rangle_n + \sum_{i=0}^{n-1}\sum_{j=n+2}^{m_2} \bar{\eta}_i \xi_j \langle \hat{p}_i, \hat{p}_j\rangle_n \\ &+ \sum_{i=n+2}^{m_1}\sum_{\substack{j=0\\ j\ne n}}^{m_2} \bar{\eta}_i \xi_j \langle \hat{p}_i, \hat{p}_j\rangle_n.\end{aligned} \tag{4.4}$$

Similarly, replacing the quadratic form $\langle\cdot,\cdot\rangle$ by $[\cdot,\cdot]_{n+1}$ in (4.2), and using the facts that $[\hat{p}_i, \hat{p}_j]_{n+1} = \langle \hat{p}_i, \hat{p}_j\rangle$ for $0 \le i, j < n$, $[\hat{p}_n, \hat{p}_j]_{n+1} = 0$ for $0 \le j < n$ as well as for $j = n+1$, $[\hat{p}_n, \hat{p}_n]_{n+1} = 2$, and $[\hat{p}_{n+1}, \hat{p}_j]_{n+1} = -\langle \hat{p}_{n+1}, \hat{p}_j\rangle_n$ for $0 \le j \le n$, yields

$$[f,g]_{n+1} = \sum_{i=0}^{n-1} \bar{\eta}_i \xi_i + 2\bar{\eta}_n \xi_n - \sum_{i=0}^{n-1} (\bar{\eta}_i \xi_{n+1} \langle \hat{p}_i, \hat{p}_{n+1}\rangle_n + \bar{\eta}_{n+1}\xi_i \langle \hat{p}_{n+1}, \hat{p}_i\rangle_n)$$

$$+\sum_{j=n+1}^{m_2}\bar{\eta}_{n+1}\xi_j\,[\hat{p}_{n+1},\hat{p}_j]_{n+1}+\sum_{i=0}^{n}\sum_{j=n+2}^{m_2}\bar{\eta}_i\xi_j\,[\hat{p}_i,\hat{p}_j]_{n+1}$$
$$+\sum_{i=n+2}^{m_1}\sum_{j=0}^{m_2}\bar{\eta}_i\xi_j\,[\hat{p}_i,\hat{p}_j]_{n+1}\,. \tag{4.5}$$

Let

$$\psi_n := \sum_{i=0}^{n-1}(\bar{\eta}_i\xi_{n+1}\langle\hat{p}_i,\hat{p}_{n+1}\rangle_n+\bar{\eta}_{n+1}\xi_i\langle\hat{p}_{n+1},\hat{p}_i\rangle_n)-\bar{\eta}_n\xi_n.$$

Combining (4.3) with (4.4), and (4.3) with (4.5), yields

$$\begin{aligned}\langle f,g\rangle_n &= \langle f,g\rangle+\psi_n+\delta_n,\\ [f,g]_{n+1} &= \langle f,g\rangle-\psi_n+\tilde{\delta}_n,\end{aligned}$$

respectively, where $\delta_n,\tilde{\delta}_n\in\mathbb{C}$ converge to zero when the coefficients η_j and ξ_j, $j\geq n+1$, do.

Assume that the coefficients η_j and ξ_j for $j\geq n+1$ are of sufficiently small magnitude, so that

$$\begin{aligned}\max\{|\mathrm{Re}(\delta_n)|,|\mathrm{Re}(\tilde{\delta}_n)|\}&\leq|\mathrm{Re}(\psi_n)|,\\ \max\{|\mathrm{Im}(\delta_n)|,|\mathrm{Im}(\tilde{\delta}_n)|\}&\leq|\mathrm{Im}(\psi_n)|,\end{aligned} \tag{4.6}$$

where $\mathrm{Re}(z)$ and $\mathrm{Im}(z)$ denote the real and imaginary parts of $z\in\mathbb{C}$, respectively. Then, if $\mathrm{Re}(\psi_n)\leq 0$, we have

$$\mathrm{Re}(\langle f,g\rangle_n)\leq\mathrm{Re}(\langle f,g\rangle)\leq\mathrm{Re}([f,g]_{n+1}).$$

Similarly, if $\mathrm{Im}(\psi_n)\leq 0$, then

$$\mathrm{Im}(\langle f,g\rangle_n)\leq\mathrm{Im}(\langle f,g\rangle)\leq\mathrm{Im}([f,g]_{n+1}).$$

Conversely, the inequality $\mathrm{Re}(\psi_n)\geq 0$ implies that $\mathrm{Re}(\langle f,g\rangle_n)$ and $\mathrm{Re}([f,g]_{n+1})$ furnish upper and lower bounds, respectively, of $\mathrm{Re}(\langle f,g\rangle)$, and $\mathrm{Im}(\psi_n)\geq 0$ implies that $\mathrm{Im}(\langle f,g\rangle_n)$ and $\mathrm{Im}([f,g]_{n+1})$ are upper and lower bounds, respectively, of $\mathrm{Im}(\langle f,g\rangle)$.

We remark that it is generally not straightforward to verify whether the conditions (4.6) hold. Nevertheless, it is interesting that there are sufficient conditions for the Arnoldi and anti-Arnoldi rules to give upper and lower bounds. Moreover, for many quadratic forms (1.3) and (1.9) these quadrature rules provide upper and lower bounds. This is illustrated in section 5.

Expansions for $\langle f,g\rangle^{(r,s)}$, $\langle f,g\rangle_n^{(r,s)}$, and $[f,g]_{n+1}^{(r,s)}$ can be derived in a way similar to the expansions (4.3), (4.4), and (4.5). Let, analogously to (4.1),

$$\|v\|f(A)v_r=\|v\|f(A)\hat{p}_{r-1}(A)v=\sum_{i=0}^{m_3}\eta_i^{(r)}\hat{p}_i(A)v,$$
$$\|v\|g(A)v_s=\|v\|g(A)\hat{p}_{s-1}(A)v=\sum_{j=0}^{m_4}\xi_j^{(s)}\hat{p}_j(A)v.$$

Then, similarly to (4.2),

$$\langle f,g\rangle^{(r,s)} = \sum_{i=0}^{m_3}\sum_{j=0}^{m_4} \bar{\eta}_i^{(r)}\xi_j^{(s)}\langle \hat{p}_i,\hat{p}_j\rangle,$$

and formulas analogous to (4.3), (4.4), and (4.5) can easily be derived. They show that under suitable conditions on the coefficients $\eta_i^{(r)}$ and $\xi_j^{(s)}$, the real and imaginary parts of the quadrature rules $\langle f,g\rangle_n^{(r,s)}$ and $[f,g]_{n+1}^{(r,s)}$ bracket the real and imaginary parts, respectively, of $\langle f,g\rangle^{(r,s)}$.

We now are in a position to discuss the computation of inexpensive estimates of bilinear forms $u^*g(A)v$, where $u=\sum_{r=1}^{\ell}\beta_r v_r$ and $r<n$, considered at the end of section 1. Assume that the coefficients β_r are real. Then

$$u^*g(A)v = \sum_{r=1}^{\ell}\beta_r v_r^*g(A)v = \frac{1}{\|v\|}\sum_{r=1}^{\ell}\beta_r\langle 1,g\rangle^{(r,1)}. \tag{4.7}$$

We use

$$\frac{1}{\|v\|}\sum_{r=1}^{\ell}\max\left\{\beta_r\langle 1,g\rangle_n^{(r,1)},\beta_r[1,g]_{n+1}^{(r,1)}\right\} \tag{4.8}$$

as an estimate of an upper bound of $u^*g(A)v$, and

$$\frac{1}{\|v\|}\sum_{r=1}^{\ell}\min\left\{\beta_r\langle 1,g\rangle_n^{(r,1)},\beta_r[1,g]_{n+1}^{(r,1)}\right\} \tag{4.9}$$

as an estimate of a lower bound.

As another application, note that from (1.7) the error $\langle f,g\rangle-\langle f,g\rangle_n$ for $\{f,g\}\in\mathbb{W}_n$ can be computed by evaluating the right-hand side of

$$\langle f,g\rangle-\langle f,g\rangle_n = \frac{1}{2}([f,g]_{n+1}-\langle f,g\rangle_n).$$

This suggests that the quadratic form $\langle\cdot,\cdot\rangle$ can be approximated by the averaged quadrature rule

$$(f,g)_{n+1/2} := \frac{1}{2}([f,g]_{n+1}+\langle f,g\rangle_n), \tag{4.10}$$

and from (1.7),

$$(f,g)_{n+1/2} = \langle f,g\rangle \qquad \forall\{f,g\}\in\mathbb{W}_n.$$

Moreover, since $\langle\hat{p}_n,\hat{p}_n\rangle=1$, $\langle\hat{p}_n,\hat{p}_n\rangle_n=0$, and $[\hat{p}_n,\hat{p}_n]=2$, the averaged quadrature rule $(\cdot,\cdot)_{n+1/2}$ takes on the same values as $\langle\cdot,\cdot\rangle$ for a larger class of functions than the quadrature rules (1.5) and (1.6). The expansion for the averaged quadrature rule,

$$\begin{aligned}(f,g)_{n+1/2} = &\sum_{i=0}^{n}\bar{\eta}_i\xi_i + \sum_{j=n+1}^{m_2}\bar{\eta}_{n+1}\xi_j(\hat{p}_{n+1},\hat{p}_j)_{n+1/2}\\ &+\sum_{i=0}^{n}\sum_{j=n+2}^{m_2}\bar{\eta}_i\xi_j(\hat{p}_i,\hat{p}_j)_{n+1/2} + \sum_{i=n+2}^{m_1}\sum_{j=0}^{m_2}\bar{\eta}_i\xi_j(\hat{p}_i,\hat{p}_j)_{n+1/2},\end{aligned}$$

does not contain many of the low-order terms present in the expansions (4.4) and (4.5) but not in the expansion (4.3), and for many functions the averaged quadrature rule (4.10) gives higher accuracy than either quadrature rule (1.5) or (1.6). This is illustrated in section 5.

Similarly to (4.10), we also introduce the averaged quadrature rules associated with the quadrature rules (2.13) and (3.14),

$$(f,g)_{n+1/2}^{(r,s)} := \frac{1}{2}([f,g]_{n+1}^{(r,s)} + \langle f,g\rangle_n^{(r,s)}), \qquad 1 \le r,s \le n, \tag{4.11}$$

as well as an averaged rule for the inexpensive approximation of (4.7),

$$u^* g(A) v \approx \frac{1}{\|v\|} \sum_{r=1}^{\ell} \beta_r \, (1,g)_{n+1/2}^{(r,1)}, \tag{4.12}$$

where the coefficients β_j are the same as in (4.7).

5. Numerical examples. This section presents computed examples which illustrate the accuracy of the computed estimates. The computations were carried out using MATLAB 6.1 on a personal computer, i.e., with approximately 15 significant digits.

In examples below, we approximate functionals of the form

$$\mathcal{I}^{(r,s)}(g) := \|v\|^2 v_r^* g(A) v_s, \qquad 1 \le r,s \le n, \tag{5.1}$$

for several functions g and matrices A by Arnoldi, anti-Arnoldi, and averaged quadrature rules given by, in order,

$$\begin{aligned}
\mathcal{G}_n^{(r,s)}(g) &:= \|v\|^2 e_r^* g(H_n) e_s, \\
\tilde{\mathcal{G}}_{n+1}^{(r,s)}(g) &:= \|v\|^2 e_r^* g(\tilde{H}_{n+1}) e_s, \\
\mathcal{L}_{2n+1}^{(r,s)}(g) &:= \frac{1}{2}(\mathcal{G}_n^{(r,s)}(g) + \tilde{\mathcal{G}}_{n+1}^{(r,s)}(g)).
\end{aligned}$$

These quadrature rules are related to the bilinear forms (2.13), (3.14), and (4.11), respectively.

Example 5.1. Let the nonsymmetric matrix $A \in \mathbb{R}^{200\times 200}$ have uniformly distributed entries in the interval $[0, 0.01]$; we generated A with the MATLAB command $\mathsf{A = rand(200)/100}$. The initial vector $v = v_1$ has random entries and is normalized to be of unit length. We compute approximations of the functionals (5.1) for $g(t) = \exp(t)$ and $1 \le r,s \le 3$. Table 5.1 displays, for $n = 3$, the values $\mathcal{I}^{(r,s)}(g)$ computed by using the MATLAB command $\mathsf{expm(A)}$ as well as approximations of these values determined by the Arnoldi, anti-Arnoldi, and averaged quadrature rules. The $n = 3$ steps of the Arnoldi process generate a 3×3 matrix H_3. Table 5.1 is a 3×3 block matrix with each block-row corresponding to one value of the index r and each block-column to one value of the index s. The first entry of each block shows the value $\mathcal{I}^{(r,s)}(g)$, and the second and third entries show the approximations determined by the Arnoldi quadrature rule $\mathcal{G}_n^{(r,s)}(g)$ and the anti-Arnoldi quadrature rule $\tilde{\mathcal{G}}_{n+1}^{(r,s)}(g)$, respectively. The last entry of each block displays the approximation obtained with the averaged rule $\mathcal{L}_{2n+1}^{(r,s)}(g)$. The values $\mathcal{I}^{(r,s)}(g)$ can be seen to lie between the approximations determined by the Arnoldi and anti-Arnoldi quadrature

TABLE 5.1
Example 5.1*:* $g(t) = \exp(t)$*,* $A \in \mathbb{R}^{200\times 200}$ *nonsymmetric random matrix,* $n = 3$*.*

	$s=1$	$s=2$	$s=3$
$\mathcal{I}^{(1,s)}(g)$	2.2393416561	0.7638513101	0.0362450361
$\mathcal{G}_n^{(1,s)}(g)$	2.2393414349	0.7638494620	0.0361306238
$\tilde{\mathcal{G}}_{n+1}^{(1,s)}(g)$	2.2393418837	0.7638532236	0.0363646813
$\mathcal{L}_{2n+1}^{(1,s)}(g)$	2.2393416593	0.7638513428	0.0362476525
$\mathcal{I}^{(2,s)}(g)$	0.7647795061	1.4693731134	0.0190982332
$\mathcal{G}_n^{(2,s)}(g)$	0.7647793198	1.4693715815	0.0190056961
$\tilde{\mathcal{G}}_{n+1}^{(2,s)}(g)$	0.7647796957	1.4693746791	0.0191935089
$\mathcal{L}_{2n+1}^{(2,s)}(g)$	0.7647795077	1.4693731303	0.0190996025
$\mathcal{I}^{(3,s)}(g)$	0.0157954394	0.0590195933	0.9964079321
$\mathcal{G}_n^{(3,s)}(g)$	0.0157953250	0.0590186907	0.9963570541
$\tilde{\mathcal{G}}_{n+1}^{(3,s)}(g)$	0.0157955541	0.0590204980	0.9964589719
$\mathcal{L}_{2n+1}^{(3,s)}(g)$	0.0157954395	0.0590195944	0.9964080130

TABLE 5.2
Example 5.2*:* $g(t) = \exp(t)$*,* $A \in \mathbb{R}^{200\times 200}$ *nonsymmetric Toeplitz matrix,* $n = 5$*.*

	$s=1$	$s=2$	$s=3$	$s=4$	$s=5$
$\mathcal{I}^{(1,s)}(g)$	201.43	−90.70	−52.26	−18.16	−7.81
$\mathcal{G}_n^{(1,s)}(g)$	201.47	−90.54	−51.78	−16.93	−5.30
$\tilde{\mathcal{G}}_{n+1}^{(1,s)}(g)$	201.40	−90.81	−52.55	−18.82	−8.80
$\mathcal{L}_{2n+1}^{(1,s)}(g)$	201.44	−90.67	−52.17	−17.88	−7.05
$\mathcal{I}^{(2,s)}(g)$	95.58	10.39	−28.53	−26.92	−12.79
$\mathcal{G}_n^{(2,s)}(g)$	95.63	10.59	−27.96	−25.42	− 9.64
$\tilde{\mathcal{G}}_{n+1}^{(2,s)}(g)$	95.55	10.25	−28.94	−27.88	−14.51
$\mathcal{L}_{2n+1}^{(2,s)}(g)$	95.59	10.42	−28.45	−26.65	−12.07

rules. The averaged rule is seen to determine approximations of higher accuracy than the Arnoldi and anti-Arnoldi quadrature rules.

Example 5.2. Let $A \in \mathbb{R}^{200\times 200}$ be a nonsymmetric Toeplitz matrix with first row $[1, 2^{-2}, 3^{-2}, \ldots, 200^{-2}]$ and first column $[1, 2^{-1}, 3^{-1}, \ldots, 200^{-1}]^T$, and let $v = [1, 1, \ldots, 1]^T/\sqrt{200}$. We compute approximations of the functionals (5.1) for $g(t) = \exp(t)$, $1 \le r \le 2$ and $1 \le s \le 5$. Table 5.2 is analogous to Table 5.1 and displays the values $\mathcal{I}^{(r,s)}(g)$ computed by using the MATLAB command $\mathsf{expm(A)}$ as well as approximations of these values determined by the Arnoldi, anti-Arnoldi, and averaged quadrature rules with $n = 5$. The values $\mathcal{I}^{(r,s)}(g)$ can be seen to lie between the approximations determined by the Arnoldi and anti-Arnoldi quadrature rules. Therefore the averaged rules determine approximations of higher accuracy than the Arnoldi and anti-Arnoldi quadrature rules.

Example 5.3. Let the nonsymmetric matrix $A \in \mathbb{R}^{500\times 500}$ and initial vector $v = v_1$ be generated similarly as in Example 5.1. We compute approximations of the functionals (5.1) for $g(t) = (1 + t^2)^{-1}$. Table 5.3 shows, for $n = 4$, the values $\mathcal{I}^{(r,s)}(g)$ computed by using the MATLAB command $\mathsf{inv}(I + A^2)$ and approximations determined by the Arnoldi, anti-Arnoldi, and averaged quadrature rules. As in the examples above, the approximations determined by the Arnoldi and anti-Arnoldi quadrature rules bracket the exact values of $\mathcal{I}^{(r,s)}(g)$. The averaged quadrature rules

TABLE 5.3
Example 5.3: $g(t) = (1+t^2)^{-1}$, $A \in \mathbb{R}^{500\times 500}$ *nonsymmetric*, $n = 4$.

	$s=1$	$s=2$	$s=3$	$s=4$
$\mathcal{I}^{(1,s)}(g)$	0.6309356066	−0.2190558983	−0.0043520021	−0.0005276674
$\mathcal{G}_n^{(1,s)}(g)$	0.6309356000	−0.2190558782	−0.0043517905	−0.0005591523
$\tilde{\mathcal{G}}_{n+1}^{(1,s)}(g)$	0.6309356129	−0.2190559178	−0.0043521878	−0.0004972393
$\mathcal{L}_{2n+1}^{(1,s)}(g)$	0.6309356065	−0.2190558980	−0.0043519891	−0.0005281958
$\mathcal{I}^{(2,s)}(g)$	−0.2188599160	0.8701328956	−0.0025997873	−0.0003641850
$\mathcal{G}_n^{(2,s)}(g)$	−0.2188599109	0.8701328732	−0.0025997154	−0.0003307974
$\tilde{\mathcal{G}}_{n+1}^{(2,s)}(g)$	−0.2188599206	0.8701329188	−0.0025999568	−0.0003982020
$\mathcal{L}_{2n+1}^{(2,s)}(g)$	−0.2188599157	0.8701328960	−0.0025998361	−0.0003644997
$\mathcal{I}^{(3,s)}(g)$	−0.0064198253	−0.0037919324	0.9999829690	−0.0000783738
$\mathcal{G}_n^{(3,s)}(g)$	−0.0064198163	−0.0037919671	0.9999829347	−0.0000258878
$\tilde{\mathcal{G}}_{n+1}^{(3,s)}(g)$	−0.0064198342	−0.0037918976	0.9999829918	−0.0001309499
$\mathcal{L}_{2n+1}^{(3,s)}(g)$	−0.0064198253	−0.0037919324	0.9999829632	−0.0000784188
$\mathcal{I}^{(4,s)}(g)$	0.0001669050	−0.0006640880	0.0000195458	1.0000332437
$\mathcal{G}_n^{(4,s)}(g)$	0.0001669043	−0.0006640839	0.0000195017	1.0000272973
$\tilde{\mathcal{G}}_{n+1}^{(4,s)}(g)$	0.0001669058	−0.0006640921	0.0000195737	1.0000391773
$\mathcal{L}_{2n+1}^{(4,s)}(g)$	0.0001669051	−0.0006640880	0.0000195377	1.0000332373

TABLE 5.4
Example 5.4: $g(t) = \exp(t)$, $A \in \mathbb{C}^{200\times 200}$ *non-Hermitian*, $n = 3$, $i = \sqrt{-1}$.

	$s=1$	$s=2$	$s=3$
$\mathcal{I}^{(1,s)}(g)$	$2.262829 + 0.039314i$	$0.742972 + 0.023252i$	$0.119483 - 0.000962i$
$\mathcal{G}_n^{(1,s)}(g)$	$2.262825 + 0.039325i$	$0.742945 + 0.023334i$	$0.119019 + 0.000486i$
$\tilde{\mathcal{G}}_{n+1}^{(1,s)}(g)$	$2.262831 + 0.039304i$	$0.742992 + 0.023169i$	$0.119814 - 0.002445i$
$\mathcal{L}_{2n+1}^{(1,s)}(g)$	$2.262828 + 0.039314i$	$0.742969 + 0.023252i$	$0.119417 - 0.000979i$
$\mathcal{I}^{(2,s)}(g)$	$0.788222 + 0.011862i$	$1.441307 + 0.021074i$	$0.061573 + 0.003638i$
$\mathcal{G}_n^{(2,s)}(g)$	$0.788225 + 0.011867i$	$1.441324 + 0.021118i$	$0.061851 + 0.004413i$
$\tilde{\mathcal{G}}_{n+1}^{(2,s)}(g)$	$0.788220 + 0.011856i$	$1.441291 + 0.021029i$	$0.061327 + 0.002829i$
$\mathcal{L}_{2n+1}^{(2,s)}(g)$	$0.788222 + 0.011862i$	$1.441307 + 0.021073i$	$0.061589 + 0.003621i$
$\mathcal{I}^{(3,s)}(g)$	$0.056832 + 0.000658i$	$0.203864 + 0.001905i$	$1.009163 + 0.004721i$
$\mathcal{G}_n^{(3,s)}(g)$	$0.056835 + 0.000654i$	$0.203882 + 0.001871i$	$1.009443 + 0.004173i$
$\tilde{\mathcal{G}}_{n+1}^{(3,s)}(g)$	$0.056830 + 0.000663i$	$0.203847 + 0.001939i$	$1.008883 + 0.005261i$
$\mathcal{L}_{2n+1}^{(3,s)}(g)$	$0.056832 + 0.000658i$	$0.203864 + 0.001905i$	$1.009163 + 0.004717i$

give approximations of higher accuracy than the Arnoldi and anti-Arnoldi quadrature rules.

Example 5.4. Let the matrices $A_1, A_2 \in \mathbb{R}^{200\times 200}$ be generated similarly to the matrix A in Example 5.1 and let $A := A_1 + iA_2 \in \mathbb{C}^{200\times 200}$, where $i = \sqrt{-1}$. The initial vector $v = v_1$ has random complex entries and is of unit length. We compute approximations of the functionals (5.1) for $g(t) = \exp(t)$ and $n = 3$. Table 5.4 displays the values $\mathcal{I}^{(r,s)}(g)$ computed by using the MATLAB command `expm(A)` as well as approximations determined by the Arnoldi, anti-Arnoldi, and averaged quadrature rules. The real and imaginary parts of $\mathcal{I}^{(r,s)}(g)$ can be seen to be bracketed by the real and imaginary parts, respectively, of the approximations determined by the Arnoldi and anti-Arnoldi quadrature rules. The averaged quadrature rule yields higher

TABLE 5.5
Example 5.5: $f(t) = t^{20}$, $g(t) = \exp(t)$, $A \in \mathbb{R}^{200\times 200}$ *nonsymmetric*, $n = 3$.

	$s=1$	$s=2$	$s=3$
$\langle f,g\rangle^{(1,s)}$	1.9426441044	1.1529977481	0.0534278853
$\langle f,g\rangle_n^{(1,s)}$	1.9427711374	1.1530783050	0.0534990864
$[f,g]_{n+1}^{(1,s)}$	1.9425056337	1.1529104434	0.0533534039
$(f,g)_{n+1/2}^{(1,s)}$	1.9426383856	1.1529943742	0.0534262451
$\langle f,g\rangle_{n+1}^{(1,s)}$	1.9426383731	1.1529943662	0.0534262419
$\langle f,g\rangle^{(2,s)}$	1.1460720705	0.6802164707	0.0315200334
$\langle f,g\rangle_n^{(2,s)}$	1.1461641820	0.6802741850	0.0315625116
$[f,g]_{n+1}^{(2,s)}$	1.1459715957	0.6801538166	0.0314755768
$(f,g)_{n+1/2}^{(2,s)}$	1.1460678888	0.6802140008	0.0315190442
$\langle f,g\rangle_{n+1}^{(2,s)}$	1.1460678789	0.6802139945	0.0315190416
$\langle f,g\rangle^{(3,s)}$	0.0532727803	0.0316184501	0.0014651433
$\langle f,g\rangle_n^{(3,s)}$	0.0536498306	0.0318423794	0.0014773829
$[f,g]_{n+1}^{(3,s)}$	0.0528602607	0.0313734723	0.0014518747
$(f,g)_{n+1/2}^{(3,s)}$	0.0532550457	0.0316079258	0.0014646288
$\langle f,g\rangle_{n+1}^{(3,s)}$	0.0532549891	0.0316078912	0.0014646132

accuracy than the Arnoldi and anti-Arnoldi quadrature rules.

Example 5.5. Let the nonsymmetric matrix $A \in \mathbb{R}^{200\times 200}$ and the initial vector $v = v_1$ be generated as in Example 5.1. We compute approximations of $\langle f,g\rangle^{(r,s)}$ for $f(t) = t^{20}$, $g(t) = \exp(t)$, and $1 \le r,s \le n$. Table 5.5 shows, for $n = 3$, the values $\langle f,g\rangle^{(r,s)}$ and approximations (2.13), (3.14), and (4.11) determined by Arnoldi, anti-Arnoldi, and averaged quadrature rules, respectively. The table also displays the approximations

$$\langle f,g\rangle_{n+1}^{(r,s)} := \|v\|^2 e_r^*(f(H_{n+1}))^* g(H_{n+1})e_s, \qquad 1 \le r,s \le n.$$

Similar to the previous examples, the approximations $\langle f,g\rangle_n^{(r,s)}$ and $[f,g]_{n+1}^{(r,s)}$ determined by the Arnoldi and anti-Arnoldi quadrature rules, respectively, bracket the exact value $\langle f,g\rangle^{(r,s)}$ for each pair (r,s). The values $(f,g)_{n+1/2}^{(r,s)}$ determined by the averaged quadrature rule can be seen to be of higher accuracy than the approximations $\langle f,g\rangle_n^{(r,s)}$ and $[f,g]_{n+1}^{(r,s)}$.

The evaluations of $\langle f,g\rangle_{n+1}^{(r,s)}$ and $(f,g)_{n+1/2}^{(r,s)}$ both require the computation of $n+1$ steps of the Arnoldi process. In Table 5.5 the value $(f,g)_{n+1/2}^{(r,s)}$ furnishes an approximation of $\langle f,g\rangle^{(r,s)}$ with a smaller error than $\langle f,g\rangle_{n+1}^{(r,s)}$ for each pair (r,s). However, we remark that it is easy to find examples such that $\langle f,g\rangle_{n+1}^{(r,s)}$ is a more accurate approximation than $(f,g)_{n+1/2}^{(r,s)}$ for some pair (r,s).

Example 5.6. Let the matrix $A \in \mathbb{R}^{200\times 200}$ and the vector $v \in \mathbb{R}^{200}$ be the same as in Example 5.2 and define $u = v + Av$. We would like to determine estimates of $u^* \exp(A)v$ using (4.8) and (4.9) with $g(t) = \exp(t)$. It follows from (1.1) with $v_1 = v$ that $u = v_1 + Av_1 = (1+h_{11})v_1 + h_{21}v_2$. Thus, $\beta_1 = 1 + h_{11}$ and $\beta_2 = h_{21}$ in (4.8) and (4.9). With $n = 4$, the formulas (4.8) and (4.9) yield the values $1.4035 \cdot 10^3$ and $1.3998 \cdot 10^3$, respectively. Using the average quadrature rule (4.12) gives $1.4016 \cdot 10^3$. The exact value is, after rounding, $1.4014 \cdot 10^3$. The value obtained by the averaged

quadrature rule is slightly more accurate than the value delivered by the Arnoldi quadrature rule with $n = 5$. The computational effort required by these rules is about the same.

The above examples show how the Arnoldi and anti-Arnoldi quadrature rules can be applied to determine estimates of upper and lower bounds for certain quadratic and bilinear forms. Moreover, the examples illustrate that for many quadratic and bilinear forms, the computed estimates are upper and lower bounds.

REFERENCES

[1] D. Calvetti, G. H. Golub, and L. Reichel, *A computable error bound for matrix functionals*, J. Comput. Appl. Math., 103 (1999), pp. 301–306.

[2] D. Calvetti, G. H. Golub, and L. Reichel, *Estimation of the L-curve via Lanczos bidiagonalization*, BIT, 39 (1999), pp. 603–619.

[3] D. Calvetti, P. C. Hansen, and L. Reichel, *L-curve curvature bounds via Lanczos bidiagonalization*, Electron. Trans. Numer. Anal., 14 (2002), pp. 20–35.

[4] D. Calvetti, S. Morigi, L. Reichel, and F. Sgallari, *Tikhonov regularization and the L-curve for large discrete ill-posed problems*, J. Comput. Appl. Math., 123 (2000), pp. 423–446.

[5] D. Calvetti, S. Morigi, L. Reichel, and F. Sgallari, *Computable error bounds and estimates for the conjugate gradient method*, Numer. Algorithms, 25 (2000), pp. 75–88.

[6] D. Calvetti, L. Reichel, and F. Sgallari, *Application of anti-Gauss quadrature rules in linear algebra*, in Applications and Computation of Orthogonal Polynomials, W. Gautschi, G. H. Golub, and G. Opfer, eds., Birkhäuser-Verlag, Basel, Switzerland, 1999, pp. 41–56.

[7] R. W. Freund and M. Hochbruck, *Gauss quadratures associated with the Arnoldi process and the Lanczos algorithm*, in Linear Algebra for Large Scale and Real-Time Application, M. S. Moonen, G. H. Golub, and B. L. R. De Moor, eds., Kluwer, Dordrecht, The Netherlands, 1993, pp. 377–380.

[8] F. R. Gantmacher, *The Theory of Matrices*, Vol. 1, Chelsea, New York, 1990.

[9] G. H. Golub and G. Meurant, *Matrices, moments and quadrature*, in Numerical Analysis 1993, D. F. Griffiths and G. A. Watson, eds., Longman Scientific and Technical, Harlow, UK, 1994, pp. 105–156.

[10] G. H. Golub and G. Meurant, *Matrices, moments and quadrature* II*: How to compute the norm of the error in iterative methods*, BIT, 37 (1997), pp. 687–705.

[11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[12] M. Hochbruck and C. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 34 (1997), pp. 1911–1925.

[13] L. Knizhnerman, *Calculation of functions of unsymmetric matrices using Arnoldi's method*, Comput. Math. Math. Phys., 31 (1991), pp. 1–9.

[14] D. P. Laurie, *Anti-Gaussian quadrature formulas*, Math. Comp., 65 (1996), pp. 739–747.

[15] G. Meurant, *The computation of bounds for the norm of the error in the conjugate gradient algorithm*, Numer. Algorithms, 16 (1997), pp. 77–87.

[16] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS, Boston, 1996.

[17] P. E. Saylor and D. C. Smolarski, *Why Gaussian quadrature in the complex plane?*, Numer. Algorithms, 26 (2001), pp. 251–280.

[18] P. E. Saylor and D. C. Smolarski, *Addendum to: "Why Gaussian quadrature in the complex plane?,"* Numer. Algorithms, 27 (2001), pp. 215–217.

[19] E. D. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, Springer-Verlag, New York, 1990.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 3, pp. 782–809

CONVERGENCE OF THE ISOMETRIC ARNOLDI PROCESS*

S. HELSEN†, A. B. J. KUIJLAARS†, AND M. VAN BAREL‡

Abstract. It is well known that the performance of eigenvalue algorithms such as the Lanczos and the Arnoldi methods depends on the distribution of eigenvalues. Under fairly general assumptions we characterize the region of good convergence for the isometric Arnoldi process. We also determine bounds for the rate of convergence and we prove sharpness of these bounds. The distribution of isometric Ritz values is obtained as the minimizer of an extremal problem. We use techniques from logarithmic potential theory in proving these results.

Key words. isometric Arnoldi process, Ritz values, equilibrium distribution, potential theory

AMS subject classifications. 15A18, 31A05, 31A15, 65F15

DOI. 10.1137/S0895479803438201

1. Introduction.

Unitary eigenvalue problems arise in a number of different fields, for example, signal processing and trigonometric approximation problems (for references, see [10]). There exist numerical methods specifically designed to solve such eigenvalue problems. In this article we examine the convergence of one such method: the isometric Arnoldi process (IAP), which was introduced by Gragg [15]. Recently, Stewart proved numerical stability of a variant in [25]. Other useful references include [9, 16].

The Arnoldi iteration method is a very popular method for computing some eigenvalues of a matrix. For a unitary matrix $U \in \mathbb{C}^{N\times N}$, the method can be adapted to exploit the structure. Here we give an outline of the method. An orthonormal basis $q_1, q_2, \ldots, q_N$ is created for $\mathbb{C}^N$ based on a Gram–Schmidt orthogonalization of the vectors $b, Ub, U^2b, \ldots, U^{N-1}b$ for some starting vector $b \in \mathbb{C}^N$. If Q is the unitary matrix with the q_j as its columns, we get $UQ = QH$ for some unitary Hessenberg matrix H, which necessarily has the same eigenvalues as U. The Arnoldi idea is to look at the $n \times n$ leading principal submatrix H_n of H (for some $n \leqslant N$) and to compute the eigenvalues of H_n. It is hoped that some of these eigenvalues are good approximants to some of the eigenvalues of U. If the required eigenvalues are indeed approximated and if $n \ll N$, then operating on H_n instead of H can save a considerable amount of computing time.

The matrix H_n is not unitary anymore, except in cases of "lucky breakdown." Ignoring such cases, we have that all eigenvalues are strictly *inside* the unit circle.

*Received by the editors November 27, 2003; accepted for publication (in revised form) by A. J. Wathen June 28, 2004; published electronically April 8, 2005. This research was partially supported by the Research Council K.U. Leuven, project OT/00/16 (SLAP: Structured Linear Algebra Package); by the Fund for Scientific Research–Flanders (Belgium), projects G.0078.01 (SMA: Structured Matrices and their Applications), G.0176.02 (ANCILA: Asymptotic aNalysis of the Convergence behavior of Iterative methods in numerical Linear Algebra), and G.0455.04 (RHPH: Riemann–Hilbert problems, random matrices, and Padé–Hermite approximation); 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/26-3/43820.html

†Department of Mathematics, Katholieke Universiteit Leuven, Celestijnenlaan 200B, 3001 Leuven, Belgium (steff@wis.kuleuven.ac.be, arno@wis.kuleuven.ac.be).

‡Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium (Marc.VanBarel@cs.kuleuven.ac.be).

The numbers we want to calculate are *on* the unit circle, so it is natural to take the approximants also on the unit circle. To this end we modify the matrix H_n. To make it a unitary matrix, it suffices to rescale the last column. Then we take the eigenvalues of the modified submatrices as approximants. This is the basic idea of the IAP. In actual implementations of the IAP, the computations are done implicitly and involve only the Schur parameters $(\gamma_n)_n$ that are associated with a unitary Hessenberg matrix.

For the convergence of the IAP, it is important to know in what sense the approximation of eigenvalues takes place and which eigenvalues are well approximated. We will consider this question from the point of view of logarithmic potential theory. Polynomial minimization problems provide the connection between Krylov subspace methods in numerical linear algebra and potential theory, which is clearly explained by Driscoll, Toh, and Trefethen [12]. See also [26, p. 279], where one finds the rule of thumb that the Lanczos iteration tends to converge to eigenvalues in regions of "too little charge" for an equilibrium distribution. This rule of thumb for the Lanczos method was made more precise in [5, 18]. It is the aim of this paper to apply similar ideas to the IAP.

Note that potential theory was also used in recent papers [4, 6, 7, 8, 24] for the convergence analysis of other iterative methods in numerical linear algebra.

The rest of the paper is organized as follows. In the next section we state our main results. Then we collect the properties of unitary Hessenberg matrices and paraorthogonal polynomials that we need for our purposes. In particular we mention a polynomial minimization problem, which is crucial for the link to potential theory. We have not seen this minimization problem in the literature before, but it may be known to specialists in the field. Section 4 contains the proofs of the main results. In the last section we will discuss some numerical experiments that illustrate our theoretical results.

2. Statement of results. The results we obtain will be of an asymptotic nature. We do not investigate the eigenvalues of a single unitary matrix U, but instead we look at a sequence of unitary matrices $(U_N)_N$, with $U_N \in \mathbb{C}^{N\times N}$. This setting reflects, for example, the discretization of a continuous problem with decreasing mesh size. The eigenvalues and orthonormal eigenvectors of U_N are denoted by $\{\lambda_{k,N}\}_{k=1}^N$ and $\{v_{k,N}\}_{k=1}^N$, respectively. We also take a unit starting vector $b_N \in \mathbb{C}^N$ for every N. For our results, we have to impose a number of mild conditions on the sequence of matrices.

In the conditions, and also in the rest of the paper, the logarithmic potential U^μ of a measure μ appears. This is the function

$$U^\mu(z) = \int \log \frac{1}{|z-z'|}\, \mathrm{d}\mu(z'),$$

which is a harmonic function outside the support of μ. The logarithmic potential U^μ may take the value $-\infty$. Further, δ_λ denotes the Dirac point mass in λ and $\|\cdot\|$ denotes the Euclidian two-norm of a vector. The unit circle in the complex plane is denoted by $\mathbb{T}$.

Conditions 2.1.

(1) There exists a probability measure σ on $\mathbb{T}$ whose logarithmic potential U^σ is

real valued and continuous, such that

$$\lim_{N\to\infty} \frac{1}{N} \sum_{j=1}^{N} \delta_{\lambda_{j,N}} = \sigma. \tag{2.1}$$

(2) For every $\varepsilon > 0$ there exists a $\delta \in (0,1)$ so that for all sufficiently large N and for all $k \leqslant N$

$$\prod_{\substack{j=1 \\ 0<|\lambda_{j,N}-\lambda_{k,N}|<\delta}}^{N} |\lambda_{j,N} - \lambda_{k,N}| > \mathrm{e}^{-N\varepsilon}. \tag{2.2}$$

(3) For every N, we have that $\|b_N\| = 1$ and

$$\lim_{N\to\infty} \Big(\min_{1\leqslant k\leqslant N} |\langle b_N, v_{k,N}\rangle| \Big)^{1/N} = 1. \tag{2.3}$$

The limit in (2.1) is in the sense of weak*-convergence of measures. In this paper convergence of measures will always be in the weak*-sense, i.e., if ν and ν_n are Borel probability measures on $\mathbb{T}$, then $\nu_n \to \nu$ if and only if

$$\int f \,\mathrm{d}\nu_n \to \int f \,\mathrm{d}\nu$$

for every continuous function f on $\mathbb{T}$. Thus the first condition states that the eigenvalues have a limiting distribution σ. The condition that U^σ is continuous and real valued (and so does not take the value $-\infty$) is a regularity condition on σ. It is satisfied, for example, if σ has a bounded density with respect to the Lebesgue measure on $\mathbb{T}$. The second condition is a technical one that prevents the eigenvalues from being too close to each other. Beckermann [5, Lemma 2.4(a)] proved that under Condition 2.1(1), Condition 2.1(2) is equivalent with the following.

(2b) For all sequences $(k_N)_N$ with $k_N \in \{1,\dots,N\}$ such that $\lim_{N\to\infty} \lambda_{k_N,N} = \lambda$ for some λ, we have

$$\lim_{N\to\infty} \frac{1}{N} \sum_{\substack{j=1 \\ j\neq k_N}}^{N} \log|\lambda_{k_N,N} - \lambda_{j,N}| = \int \log|\lambda - z| \,\mathrm{d}\sigma(z). \tag{2.2b}$$

A discussion about this condition can be found in [19]. The third condition imposes that the starting vectors are sufficiently random, i.e., their eigenvector components are not exponentially small. Since the numbers $|\langle b_N, v_{j,N}\rangle|$ will be used frequently, we introduce a shorter notation:

$$w_{j,N} := |\langle b_N, v_{j,N}\rangle|. \tag{2.4}$$

For every N we consider the IAP on U_N with starting vector b_N. Iteratively, an orthonormal basis is created for the Krylov subspaces

$$\mathcal{K}_{n,N} = \operatorname{span}\{b_N, U_N b_N, U_N^2 b_N, \dots, U_N^{n-1} b_N\}.$$

If we compute a basis for whole $\mathbb{C}^N$ in this way, U_N is represented by a Hessenberg matrix in this basis. The $n \times n$ principal left upper block $H_{n,N}$ of this matrix is the

representation of the orthogonal projection of U_N onto $\mathcal{K}_{n,N}$. By modifying the last column of $H_{n,N}$ we can obtain a unitary Hessenberg matrix $\tilde{H}_{n,N}$. The modification depends on a unimodular constant $\rho_{n,N}$; see also section 3. The precise value of $\rho_{n,N}$ is not important to our results, and we will not indicate the dependence on $\rho_{n,N}$ in our notation. Let

$$\psi_{n,N}(z) = \det(zI_n - \tilde{H}_{n,N}), \tag{2.5}$$

where I_n denotes the $n \times n$ identity matrix and let $\theta_{1,n,N}, \theta_{2,n,N}, \ldots, \theta_{n,n,N}$ be the zeros of $\psi_{n,N}$. We call these numbers the *Ritz values for the IAP* or the *isometric Ritz values*. Since they are the eigenvalues of $\tilde{H}_{n,N}$, which is a unitary matrix, the isometric Ritz values are on the unit circle. We take the eigenvalues of the matrices U_N and the isometric Ritz values to be numbered counterclockwise, but we do not specify a starting point. We also take $\lambda_{0,N} := \lambda_{N,N}$ and $\theta_{0,n,N} := \theta_{n,n,N}$.

In section 3 (see Proposition 3.4 below) we will prove that the isometric Ritz values are separated by the eigenvalues, by which we mean that on the open arc between two consecutive isometric Ritz values there is at least one eigenvalue, or put differently, on the closed arc between any two consecutive eigenvalues there is at most one isometric Ritz value.

We consider the convergence of isometric Ritz values along ray sequences, i.e., we let N approach infinity, and with it also n, in such a fashion that $n/N \to t$ for some $t \in (0,1)$. If we consider the points (N,n) in a triangular array, then the convergence is taken along a sequence of (N,n) values that are asymptotic to a line with slope t in the N-n plane. We denote a limit in this sense by $\lim_{n,N\to\infty, n/N\to t}$.

THEOREM 2.2. *Let (U_N) and (b_N) be such that Conditions 2.1 hold. Then for every $t \in (0,1)$, there exists a Borel probability measure μ_t, depending only on t and σ, such that*

$$\lim_{\substack{n,N\to\infty \\ n/N\to t}} \frac{1}{n} \sum_{j=1}^{n} \delta_{\theta_{j,n,N}} = \mu_t \tag{2.6}$$

and a real constant F_t such that

$$\lim_{\substack{n,N\to\infty \\ n/N\to t}} \|\psi_{n,N}(U_N) b_N\|^{1/n} = \exp(-F_t). \tag{2.7}$$

The measure μ_t satisfies

$$0 \leqslant t\mu_t \leqslant \sigma, \qquad \int \mathrm{d}\mu_t = 1 \tag{2.8}$$

and minimizes the logarithmic energy

$$I(\mu) = \iint \log \frac{1}{|z - z'|} \, \mathrm{d}\mu(z) \mathrm{d}\mu(z') \tag{2.9}$$

among all measures μ satisfying $0 \leqslant t\mu \leqslant \sigma$ and $\int \mathrm{d}\mu = 1$. The logarithmic potential U^{μ_t} of μ_t is a continuous function on $\mathbb{C}$, and the constant F_t is such that

$$\begin{cases} U^{\mu_t}(z) = F_t & \text{for } z \in \operatorname{supp}(\sigma - t\mu_t), \\ U^{\mu_t}(z) \leqslant F_t & \text{for } z \in \mathbb{C}. \end{cases} \tag{2.10}$$

Furthermore, the relations (2.8) *and* (2.10) *characterize the pair* (μ_t, F_t).

This theorem tells us that the isometric Ritz values have a limiting distribution μ_t if we let $n, N \to \infty$ in such a way that $n/N \to t$. The measure μ_t is the minimizer of the logarithmic energy (2.9) under the constraints (2.8). Conditions (2.10) are the Euler–Lagrange variational conditions for this minimization problem and together with (2.8) they also characterize μ_t.

The next theorem shows that in a certain region the isometric Ritz values converge exponentially fast to eigenvalues.

THEOREM 2.3. *Let* (U_N) *and* (b_N) *be such that Conditions* 2.1 *hold and let* F_t *be as in Theorem* 2.2. *Then we have, for every* $t \in (0,1)$,

$$\limsup_{\substack{n,N\to\infty \\ n/N\to t}} \min_j |\lambda_{k_N,N} - \theta_{j,n,N}|^{1/n} \leqslant \exp\bigl(U^{\mu_t}(\lambda) - F_t\bigr) \tag{2.11}$$

for every sequence of indices (k_N) *with* $1 \leqslant k_N \leqslant N$, *such that* $(\lambda_{k_N,N})_N$ *converges to* $\lambda \in \mathbb{T}$.

We define the set

$$\Lambda(t,\sigma) := \{\lambda \in \mathbb{T} \mid U^{\mu_t}(\lambda) < F_t\}.$$

This is the region of good convergence of the IAP in the regime we are considering. Inside this set, the right-hand side of (2.11) is strictly less than 1, which indicates that for large N, an eigenvalue $\lambda_{k_N,N}$ of U_N in $\Lambda(t,\sigma)$ is approximated by an isometric Ritz value at a geometric rate. Outside $\Lambda(t,\sigma)$, the right-hand side is just one and then no convergence can be guaranteed.

In the next theorem, we will show that the convergence rate is actually twice as big, except for perhaps one eigenvalue. It is also proven that this convergence bound is sharp. In the theorem there will appear "exceptional indices": the sharper convergence rate will hold for all indices except for these "exceptional indices."

THEOREM 2.4. *Let* (U_N) *and* (b_N) *be such that Conditions* 2.1 *hold, let* F_t *be as in Theorem* 2.2, *and let* $\lambda \in \Lambda(t,\sigma)$. *Then for every* N, *there exists at most one index* $k_N^*(\lambda) \in \{1,2,\dots,N\}$ *such that the following holds. If* (k_N) *is a sequence of indices with* $1 \leqslant k_N \leqslant N$ *and* $k_N \neq k_N^*(\lambda)$ *for every* N *large enough, such that* $(\lambda_{k_N,N})_N$ *converges to* λ, *then we have*

$$\lim_{\substack{n,N\to\infty \\ n/N\to t}} \min_j |\lambda_{k_N,N} - \theta_{j,n,N}|^{1/n} = \exp\Bigl(2\bigl(U^{\mu_t}(\lambda) - F_t\bigr)\Bigr). \tag{2.12}$$

Remark 2.5. The fact that the convergence rate can be doubled was first realized by Beckermann [5] in the context of the convergence of the Lanczos method. He also introduced the exceptional indices. The proof of Theorem 2.4 is based on the proof of [5, Theorem 2.1], but we have streamlined some of the arguments; see section 4.4 below.

Remark 2.6. It is possible to prove the inequality

$$\limsup_{\substack{n,N\to\infty \\ n/N\to t}} \min_j |\lambda_{k_N,N} - \theta_{j,n,N}|^{1/n} \leqslant \exp\Bigl(2\bigl(U^{\mu_t}(\lambda) - F_t\bigr)\Bigr)$$

under weaker conditions; see [5].

There are two types of possible exceptional behavior for the index $k_N^*(\lambda)$ in Theorem 2.4, namely,

$$\min_j |\lambda_{k_N^*(\lambda),N} - \theta_{j,n,N}|^{1/n} \gg \exp\Big(2\big(U^{\mu_t}(\lambda) - F_t\big)\Big), \tag{2.13a}$$

$$\min_j |\lambda_{k_N^*(\lambda),N} - \theta_{j,n,N}|^{1/n} \ll \exp\Big(2\big(U^{\mu_t}(\lambda) - F_t\big)\Big). \tag{2.13b}$$

According to Theorem 2.4 at most one of them can occur for a fixed N. So we have three possible situations: no exception, exception (2.13a), or exception (2.13b). Which situation occurs will depend on the choice of parameter $\rho_{n,N}$. To show what happens, we will make a classification of the relative positioning of the isometric Ritz values and the eigenvalues in a closed arc $I \subset \Lambda(t,\sigma)$.

It will be shown in Proposition 3.4 that the isometric Ritz values are separated by the eigenvalues, and (2.11) tells us that each eigenvalue in $\Lambda(t,\sigma)$ is approximated at an exponential rate. Since the gaps between eigenvalues are *not* exponentially small (see Lemma 4.2), each Ritz value can be close to a single eigenvalue only, if N is large enough. From this information we can make a complete classification of the relative positions of eigenvalues and isometric Ritz values on the arc $I \subset \Lambda(t,\sigma)$.

- Case 1: Each eigenvalue in I is close to exactly one isometric Ritz value and the isometric Ritz value follows closely after the eigenvalue (when looking in the counterclockwise direction).
- Case 2: One eigenvalue in I is close to two isometric Ritz values, one on each side of it.
- Case 3: One isometric Ritz value in I is not close to an eigenvalue.
- Case 4: Each eigenvalue in I is close to exactly one isometric Ritz value and the isometric Ritz value precedes the eigenvalue (when looking in the counterclockwise direction).
- Case 5: One isometric Ritz value in I coincides with an eigenvalue.
- Case 6: One arc between two consecutive eigenvalues in I contains no isometric Ritz values.

The six different cases are illustrated in Figure 2.1, are mutually exclusive, and cover all possibilities. In [2] and [3, Theorem 2.12] one can find a similar description of the zeros of discrete orthogonal polynomials on the real line.

Recall that the IAP depends on the choice of a unimodular constant $\rho_{n,N}$. If we move $\rho_{n,N}$ around the unit circle in the counterclockwise direction, the isometric Ritz values also move in the counterclockwise direction, as shown in Figure 2.1. If we start in Case 1, no isometric Ritz value can leave "its" eigenvalue until an extra isometric Ritz value enters the arc I from the right, then we are in Case 2. Next, one isometric Ritz value is free to move away from its eigenvalue, and we pass via Case 3 to Case 2 again. This process is shown in parts (a)–(d) of Figure 2.1. Continuing this way, we see that the eigenvalue that is well approximated by two isometric Ritz values "moves" through I, until it drops off and we reach Case 4 (part (h) of Figure 2.1). We stay in Case 4 until the left-most isometric Ritz value reaches "its" eigenvalue. Then one isometric Ritz value exactly coincides with an eigenvalue and we are in Case 5. The left-most isometric Ritz value then passes the eigenvalue and we are in Case 6, where there are two consecutive eigenvalues without an isometric Ritz value on the arc between them. We refer to this arc as a gap. The gap moves to the right as shown in parts (j)–(n) of Figure 2.1, until we reach Case 1 again; see part (p).

Now we turn to the exceptional cases. In Cases 1, 3, and 4, there are no exceptions. The exception (2.13a) may occur in Case 2. In Case 2 there are two isometric Ritz

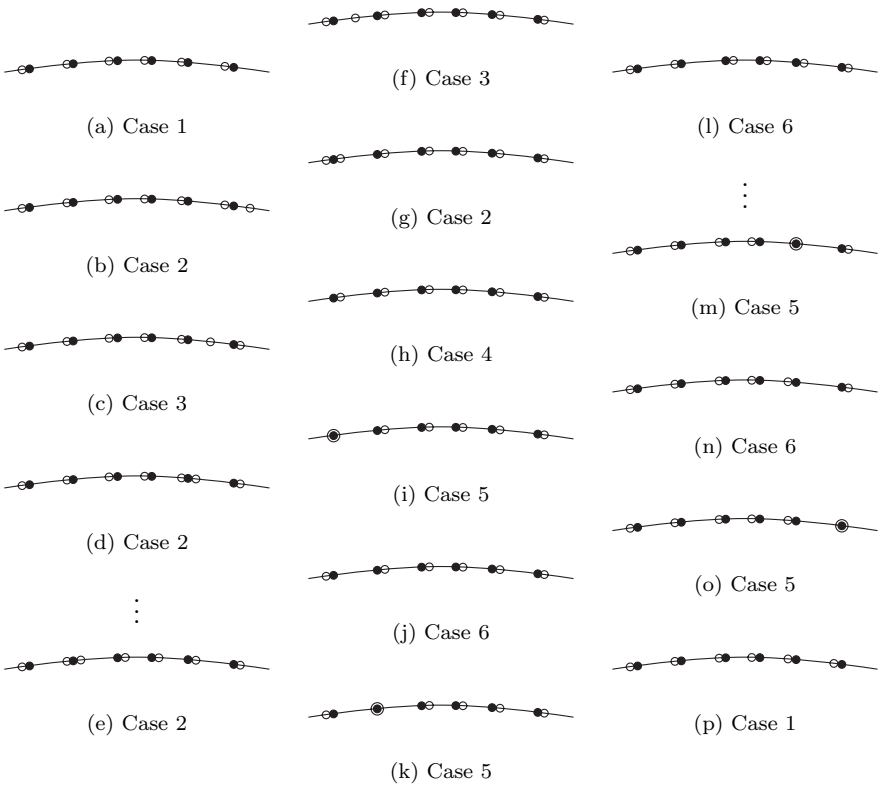


FIG. 2.1. *The evolution of the isometric Ritz values in a closed arc* $I \subset \Lambda(t, \sigma)$ *when* $\rho_{n,N}$ *moves counterclockwise around* $\mathbb{T}$. *The full dots are the eigenvalues and the open circles are the isometric Ritz values. The possibilities of their location are the following.*
Case 1: *An isometric Ritz value follows after each eigenvalue at close distance.*
Case 2: *Two isometric Ritz values are close to the same eigenvalue.*
Case 3: *One isometric Ritz value is not close to any eigenvalue.*
Case 4: *An isometric Ritz value precedes each eigenvalue at close distance.*
Case 5: *One isometric Ritz value coincides with an eigenvalue.*
Case 6: *One arc between two eigenvalues contains no isometric Ritz values.*

values close to the same eigenvalue. In this case the doubling of the exponent in (2.12) need not take place.

In Cases 5 and 6 the exception (2.13b) appears. This is clear if an eigenvalue and an isometric Ritz value coincide, which corresponds to Case 5. In Case 6 there is a gap and this case arises out of Case 5 after a small perturbation of the parameter. For a sufficiently small perturbation, the isometric Ritz value is still closer to the eigenvalue than predicted by (2.12). So in Case 6 there may be one eigenvalue around the gap with an isometric Ritz value that is too close to it. This eigenvalue corresponds to the exceptional index. It may be somewhat surprising that only one of the eigenvalues around the gap may be an exception while the other one is not.

Remark 2.7. Theorems 2.3 and 2.4 are clearly of an asymptotic nature. They express that eigenvalues in the set $\Lambda(t, \sigma)$ are well approximated by isometric Ritz values, provided n and N are large enough. In certain situations one might be dealing with a single unitary matrix and in such a case it is not clear whether the matrix is

large enough or not. Actually, our methods do not provide a framework for looking at a single matrix. Indeed, a basic assumption is that the eigenvalues of the matrix are distributed according to a measure σ (see Condition 2.1(1)), and this notion does not make sense for a fixed single matrix. In such a case, our results can only give an indication of the convergence behavior of the IAP.

On the other hand, it might happen that the unitary matrix is naturally embedded in a sequence of unitary matrices if it arises from a discretization of a physical system. This is, for example, the case in the signal processing context discussed in [9]. Then it is reasonable to assume that the sequence of matrices has a limiting eigenvalue distribution as in Condition 2.1(1), and our results apply to the full sequence. Again, our results do not apply to an individual matrix. However, our experience shows that if the matrix comes from a sequence with a limiting eigenvalue distribution, then the convergence behavior predicted by the theory can already be observed for matrices of moderate size (say 500×500). Therefore we believe our results can be of help to understand the convergence behavior of IAP, also when applied to matrices of a fixed finite size.

3. Unitary Hessenberg matrices and para-orthogonal polynomials. In this section we collect a number of results that can be found in various sources and we put them in a form that is convenient for our purposes. The size N is fixed throughout this section and will not be indicated in the notation.

We consider a unitary matrix U of size $N \times N$ with simple eigenvalues $\lambda_1, \ldots, \lambda_N$ and corresponding normalized eigenvectors $v_1, \ldots, v_N$. We also consider a unit starting vector $b \in \mathbb{C}^N$ with a nonzero component in the direction of every eigenvector. We define a measure

$$\nu = \sum_{j=1}^{N} w_j^2 \delta_{\lambda_j} = \sum_{j=1}^{N} |\langle b, v_j \rangle|^2 \delta_{\lambda_j}.$$

Since b is a unit vector and the v_j form an orthonormal basis of $\mathbb{C}^N$, we have that

$$\int \mathrm{d}\nu = \sum_{j=1}^{N} |\langle b, v_j \rangle|^2 = \|b\|^2 = 1,$$

so that ν is a discrete probability measure supported on the eigenvalues λ_j.

LEMMA 3.1. *For every function $f : \mathbb{T} \to \mathbb{C}$, we have*

$$\|f(U)b\|^2 = \int |f|^2 \, \mathrm{d}\nu.$$

Proof. Let V be the unitary matrix with the v_j as columns and let Λ be the diagonal matrix with the λ_j on the diagonal, so $U = V\Lambda V^*$ is the eigenvalue decomposition of U. Then $f(U) = Vf(\Lambda)V^*$ and, since V is unitary,

$$\|f(U)b\| = \|Vf(\Lambda)V^*b\| = \|f(\Lambda)V^*b\|.$$

Now $f(\Lambda)$ is a diagonal matrix with $f(\lambda_j)$ on the diagonal and V^*b is a vector whose jth component is $v_j^*b = \langle b, v_j \rangle$. Hence

$$\|f(U)b\|^2 = \sum_{j=1}^{N} |f(\lambda_j)\langle b, v_j \rangle|^2 = \int |f|^2 \, \mathrm{d}\nu,$$

which proves the lemma. □

If carried out to the end, the IAP transforms the unitary matrix U to the $N \times N$ unitary upper Hessenberg matrix H,

$$H = \begin{bmatrix} h_{11} & h_{12} & & \cdots & h_{1N} \\ h_{21} & h_{22} & & & \\ & h_{32} & \ddots & & \vdots \\ & & \ddots & & \\ & & & h_{N,N-1} & h_{NN} \end{bmatrix},$$

with real and positive subdiagonal elements $h_{j+1,j} > 0$. The eigenvalues of H and U are the same. The principal leading submatrix of size $n \times n$ will be denoted by H_n. The matrices H_n, $n < N$, are not unitary, since the norm of the last column of H_n is strictly less than one. We define the characteristic polynomials

$$\phi_n(z) = \det(zI_n - H_n).$$

LEMMA 3.2. *The polynomial ϕ_n is the monic polynomial of degree n that is orthogonal with respect to ν.*

Proof. We define polynomials φ_n, $n = 0, \ldots, N$, recursively by $\varphi_0(z) \equiv 1$ and

$$z\varphi_k(z) = \sum_{j=0}^{k+1} h_{j+1,k+1}\varphi_j(z) \qquad \text{for } k = 0, \ldots, N-1, \tag{3.1}$$

where we have put (somewhat arbitrarily) $h_{N+1,N} = 1$. Then we have, for $n \leqslant N$,

$$\begin{aligned} &\begin{bmatrix} \varphi_0(z) & \varphi_1(z) & \cdots & \varphi_{n-1}(z) \end{bmatrix} H_n \\ &\qquad = z \begin{bmatrix} \varphi_0(z) & \cdots & \varphi_{n-1}(z) \end{bmatrix} - \begin{bmatrix} 0 & \cdots & 0 & h_{n+1,n}\varphi_n(z) \end{bmatrix}. \end{aligned} \tag{3.2}$$

It follows from (3.2) that every zero of φ_n is an eigenvalue of H_n. This shows that φ_n is a multiple of ϕ_n provided that the zeros of φ_n are simple. If z_0 is a zero of φ_n of multiplicity m, then taking j derivatives of (3.2) and putting $z = z_0$, we get, for every $j = 1, \ldots, m-1$,

$$\begin{aligned} &\begin{bmatrix} \varphi_0^{(j)}(z_0) & \varphi_1^{(j)}(z_0) \cdots & \varphi_{n-1}^{(j)}(z_0) \end{bmatrix} H_n \\ &\qquad = z_0 \begin{bmatrix} \varphi_0^{(j)}(z_0) & \cdots & \varphi_{n-1}^{(j)}(z_0) \end{bmatrix} + j \begin{bmatrix} \varphi_0^{(j-1)}(z_0) & \cdots & \varphi_{n-1}^{(j-1)}(z_0) \end{bmatrix}. \end{aligned} \tag{3.3}$$

Thus the vectors

$$\frac{1}{j!} \begin{bmatrix} \varphi_0^{(j)}(z_0) & \varphi_1^{(j)}(z_0) & \cdots & \varphi_{n-1}^{(j)}(z_0) \end{bmatrix}, \qquad j = 0, 1, \ldots, m-1,$$

are a left Jordan chain for H_n of length m, which means that z_0 is a zero of $\phi_n(z) = \det(zI - H_n)$ of multiplicity at least m. Since this holds for every zero of φ_n, we get that φ_n is a multiple of ϕ_n also in the case of multiple eigenvalues. The leading coefficient of φ_n can be computed with (3.1) and we see that

$$\varphi_n(z) = \left(\prod_{j=1}^{n} h_{j+1,j}^{-1} \right) \det(zI_n - H_n) = \left(\prod_{j=1}^{n} h_{j+1,j}^{-1} \right) \phi_n(z); \tag{3.4}$$

see also [13].

From (3.2) with $n = N$, it follows that $\begin{bmatrix}\varphi_0(\lambda_j) & \varphi_1(\lambda_j) & \cdots & \varphi_{N-1}(\lambda_j)\end{bmatrix}$ is a left eigenvector of H for the eigenvalue λ_j. Let

$$\tilde{w}_j = \|\begin{bmatrix}\varphi_0(\lambda_j) & \varphi_1(\lambda_j) & \cdots & \varphi_{N-1}(\lambda_j)\end{bmatrix}\|^{-1}$$

so that $\begin{bmatrix}\tilde{w}_j\varphi_0(\lambda_j) & \tilde{w}_j\varphi_1(\lambda_j) & \cdots & \tilde{w}_j\varphi_{N-1}(\lambda_j)\end{bmatrix}$ is a normalized eigenvector of H. Since the matrix H is unitary (hence normal) with simple spectrum, its normalized eigenvectors form an orthonormal basis of $\mathbb{C}^n$. Thus

$$S = \begin{bmatrix} \tilde{w}_1\varphi_0(\lambda_1) & \cdots & \tilde{w}_1\varphi_{N-1}(\lambda_1) \\ \vdots & \ddots & \vdots \\ \tilde{w}_N\varphi_0(\lambda_N) & \cdots & \tilde{w}_N\varphi_{N-1}(\lambda_N) \end{bmatrix}$$

is unitary. Then $S^*S = I$ and if we look at the individual matrix entries of this last expression, we find

$$\sum_{j=1}^{N} \tilde{w}_j^2 \varphi_k(\lambda_j)\varphi_l(\lambda_j) = \delta_{k,l} \qquad \text{for } k, l = 0, 1, \ldots, N-1.$$

So the polynomials φ_n are orthonormal polynomials with respect to the measure $\sum_{j=1}^{N} \tilde{w}_j^2 \delta_{\lambda_j}$, and because of (3.4) we have that the polynomials ϕ_n are the monic orthogonal polynomials with respect to this measure.

Now we show $\tilde{w}_j = |\langle b, v_j\rangle|$ for $j = 1, \ldots, N$ to complete the proof of the lemma. We know that $UQ = QH$ where Q is a unitary matrix whose first column is b. From the eigenvalue decomposition $U = V\Lambda V^*$ we get that $V^*QH = \Lambda V^*Q$, which means that v_j^*Q is a normalized left eigenvector of H for the eigenvalue λ_j. Also $\begin{bmatrix}\tilde{w}_j\varphi_0(\lambda_j) & \cdots & \tilde{w}_j\varphi_{N-1}(\lambda_j)\end{bmatrix}$ is a normalized left eigenvector with λ_j. Then the first components have the same absolute values. The first column of Q is equal to b so that the first component of v_j^*Q is equal to $v_j^*b = \langle b, v_j\rangle$. Thus we have $\tilde{w}_j = |\tilde{w}_j\phi_0(\lambda_j)| = |\langle b, v_j\rangle|$. ∎

The previous lemma connects the Arnoldi process to the theory of orthogonal polynomials and in particular to the Arnoldi minimization problem; see, for example, [26].

ARNOLDI MINIMIZATION PROBLEM. *Minimize $\|p_n(U)b\|$ among all monic polynomials p_n of degree n.*

It is a general fact that the monic polynomial ϕ_n of degree n which is orthogonal with respect to μ minimizes the $L^2(\mu)$ norm $(\int |p_n|^2 \, \mathrm{d}\mu)^{1/2}$ among all monic polynomials p_n of degree n. Because of Lemma 3.1 it is then clear that ϕ_n is the minimizer in the Arnoldi minimization problem.

We want to establish a similar minimization problem for the isometric Arnoldi process. To that end we first recall that H can be decomposed as a product of Givens reflectors [15] (see also [1]):

$$H = G_1(\gamma_1)G_2(\gamma_2)\cdots G_{N-1}(\gamma_{N-1})\tilde{G}_N(\gamma_N),$$

for some complex parameters γ_j satisfying $|\gamma_j| < 1$ for $j = 1, \ldots, N-1$ and $|\gamma_N| = 1$. The matrices $G_j(\alpha)$ are given by

$$G_j(\alpha) = \begin{bmatrix} I_{j-1} & & & \\ & -\alpha & \sqrt{1-|\alpha|^2} & \\ & \sqrt{1-|\alpha|^2} & \bar{\alpha} & \\ & & & I_{N-j-1} \end{bmatrix},$$

and $\tilde{G}_N(\gamma_N)$ is given by

$$\tilde{G}_N(\alpha) = \begin{bmatrix} I_{N-1} & \\ & -\alpha \end{bmatrix}.$$

The numbers γ_j are called the Schur parameters for the unitary Hessenberg matrix H. We use the notation $H = H(\gamma_1, \ldots, \gamma_N)$. If we define $\sigma_j := \sqrt{1-|\gamma_j|^2}$ and write out the above product, we get an explicit expression for H in terms of the Schur parameters:

$$H = \begin{bmatrix} -\gamma_1 & -\sigma_1\gamma_2 & -\sigma_1\sigma_2\gamma_3 & \cdots & -\sigma_1\cdots\sigma_{N-2}\gamma_{N-1} & -\sigma_1\cdots\sigma_{N-1}\gamma_N \\ \sigma_1 & -\bar{\gamma}_1\gamma_2 & -\bar{\gamma}_1\sigma_2\gamma_3 & & & \\ & \sigma_2 & -\bar{\gamma}_2\gamma_3 & & & \\ & & \sigma_3 & \ddots & \vdots & \vdots \\ & & & \ddots & & \\ & & & & & -\bar{\gamma}_{N-2}\sigma_{N-1}\gamma_N \\ & & & & \sigma_{N-1} & -\bar{\gamma}_{N-1}\gamma_N \end{bmatrix}.$$

From this expression for the matrix, it is easy to see that $H_n = H(\gamma_1, \ldots, \gamma_n)$. Since the matrices $G_j(\alpha)$ have determinant -1, and $\tilde{G}_n(\alpha)$ has determinant $-\alpha$, it easily follows that [14]

$$\phi_n(0) = \det(-H_n) = \gamma_n \qquad \text{for } n = 1, \ldots, N. \tag{3.5}$$

As mentioned before, the IAP modifies the matrix H_n in order to make it unitary. The only thing that needs to change is the length of the last column. To rescale that last column, we construct

$$\tilde{H}_n := H(\gamma_1, \ldots, \gamma_{n-1}, \rho_n)$$

with ρ_n a unimodular number. This transformation amounts to multiplying the last column of H_n by the number $\frac{\rho_n}{\gamma_n}$ (provided $\gamma_n \neq 0$). Note that the parameter ρ_n can be anywhere on the unit circle. The matrices $\tilde{H}_n$ do depend on the precise choice of ρ_n, but its location will not be of any importance to us, as can be seen from the theorems. As a consequence, we do not include the dependence on ρ_n in the notation.

We will need the concept of para-orthogonal polynomials. To that end, we recall their definition; see, for example, [17]. For a polynomial p of degree n, let

$$p^*(z) = z^n \overline{p(1/\bar{z})}$$

be the reciprocal polynomial. The (monic) para-orthogonal polynomials ψ_n are then defined by

$$\psi_n(z) := \frac{\phi_n(z) + \omega_n \phi_n^*(z)}{1 + \omega_n \bar{\gamma}_n}, \tag{3.6}$$

where ϕ_n is the monic orthogonal polynomial with respect to the measure ν and $\omega_n \in \mathbb{T}$. Note that in the literature the para-orthogonal polynomials are usually defined as $\phi_n + \omega_n \phi_n^*$ so that they are not monic.

We have to be careful here, since we have already defined a set of polynomials $\psi_{n,N}$ in (2.5). In fact, the two definitions are the same. More precisely, for every

$\rho_n \in \mathbb{T}$ there exists an $\omega_n \in \mathbb{T}$, and conversely for every $\omega_n \in \mathbb{T}$ there exists a $\rho_n \in \mathbb{T}$, such that

$$\psi_n(z) = \det(zI_n - \tilde{H}_n),$$

where ψ_n is defined as in (3.6). The 1-1 correspondence between ρ_n and ω_n is given by

$$\rho_n = \omega_n\left(\frac{1+\bar{\omega}_n\gamma_n}{1+\omega_n\bar{\gamma}_n}\right), \qquad \omega_n = \rho_n\left(\frac{1-\bar{\rho}_n\gamma_n}{1-\rho_n\bar{\gamma}_n}\right). \tag{3.7}$$

This is a consequence of a remark in [1] and is easily verified using the recurrence relations for the orthogonal polynomials and their reciprocals which are stated in [15].

These polynomials are called para-orthogonal since they are orthogonal with respect to ν to all polynomials of degree less than n without constant term [17], that is,

$$\int_{\mathbb{T}} \psi_n(z)\bar{z}^k \,\mathrm{d}\nu(z) = 0, \qquad k = 1, 2, \ldots, n-1. \tag{3.8}$$

It is known that the zeros of ψ_n are simple and lie on the unit circle [15, 17]. We denote them by $\theta_1, \ldots, \theta_n$. We recall that these zeros are the basis of the Gauss quadrature formula on the unit circle [17]

$$\sum_{j=1}^{n} \beta_j p(\theta_j) = \int p \,\mathrm{d}\nu, \qquad \beta_j > 0, \tag{3.9}$$

which is valid for Laurent polynomials p of degree $\leqslant n-1$ (i.e., for linear combinations of z^k with $k = -n+1, -n+2, \ldots, n-2, n-1$).

Our next result, which is the main result of this section, states that the para-orthogonal polynomials solve a minimization problem, similar to the Arnoldi minimization problem. We call it the isometric Arnoldi minimization problem. While this result may be known already, we have not seen it in the literature.

Isometric Arnoldi minimization problem. *Minimize $\|p_n(U)b\|$ among all monic polynomials p_n of degree n satisfying $p_n(0) = \rho_n$, where $\rho_n \in \mathbb{T}$ is given.*

Theorem 3.3. *The minimizer of the isometric Arnoldi minimization problem is unique and is given by the monic para-orthogonal polynomial ψ_n, where ω_n is related to ρ_n as in* (3.7).

Proof. Let ψ_n be the monic para-orthogonal polynomial of degree n with parameter $\omega_n = \rho_n\big(\frac{1-\bar{\rho}_n\gamma_n}{1-\rho_n\bar{\gamma}_n}\big)$. Using the same reasoning as the one leading to (3.5), we find $\psi_n(0) = \rho_n$.

If p_n is an arbitrary monic polynomial of degree n with $p_n(0) = \rho_n$, then $p_n - \psi_n$ is a linear combination of $z, z^2, \ldots, z^{n-1}$, so that by the para-orthogonality property (3.8) we have

$$\int \psi_n(z)\overline{(p_n(z) - \psi_n(z))} \,\mathrm{d}\nu(z) = 0.$$

Thus

$$\int \psi_n \bar{p}_n \,\mathrm{d}\nu = \int |\psi_n|^2 \,\mathrm{d}\nu.$$

This leads to

$$\int |p_n - \psi_n|^2 \, \mathrm{d}\nu = \int |p_n|^2 \, \mathrm{d}\nu - \int |\psi_n|^2 \, \mathrm{d}\nu,$$

from which we deduce that

$$\int |p_n|^2 \, \mathrm{d}\nu \geqslant \int |\psi_n|^2 \, \mathrm{d}\nu,$$

with equality if and only if $\int |p_n - \psi_n|^2 \, \mathrm{d}\nu = 0$. Since $p_n - \psi_n$ is a polynomial of degree $n-1$ and the measure ν is carried on N points, equality can hold only if $p_n = \psi_n$. Thus by Lemma 3.1

$$\|p_n(U)b\| \geqslant \|\psi_n(U)b\|$$

with equality if and only if $p_n = \psi_n$. This proves the theorem. ∎

Using Theorem 3.3 we will now prove that the zeros of the para-orthogonal polynomial ψ_n (which are on the unit circle) are separated by the eigenvalues of U.

PROPOSITION 3.4. *Let $n < N$. Then the zeros of ψ_n are separated by the eigenvalues of U.*

Proof. Let θ_1 and θ_2 be two distinct zeros of ψ_n. We have to show that there is an eigenvalue on the open arc between θ_1 and θ_2. Without loss of generality, we may restrict ourselves to the case that $\theta_1 = \mathrm{e}^{-is_0}$, $\theta_2 = \mathrm{e}^{is_0}$, where $s_0 \in (0, \pi)$. The eigenvalues are of the form $\lambda_j = \mathrm{e}^{is_j}$ with $-\pi \leqslant s_j \leqslant \pi$. Then

$$\psi_n(z) = (z - \mathrm{e}^{-is_0})(z - \mathrm{e}^{is_0}) q_{n-2}(z), \tag{3.10}$$

where q_{n-2} is a polynomial of degree $n-2$. We know from Theorem 3.3 that ψ_n minimizes

$$\|p_n(U)b\|^2 = \sum_{j=1}^{N} w_j^2 |p_n(\lambda_j)|^2,$$

among all monic polynomials of degree n with $p_n(0) = \rho_n$ (see also Lemma 3.1). For each s, we have that $(z - \mathrm{e}^{-is})(z - \mathrm{e}^{is}) q_{n-2}(z)$ is a monic polynomial with value ρ_n at $z = 0$. Thus

$$I(s) := \sum_{j=1}^{N} w_j^2 (|\lambda_j - \mathrm{e}^{-is}||\lambda_j - \mathrm{e}^{is}|)^2 |q_{n-2}(\lambda_j)|^2$$

is minimal for $s = s_0$. Observe that

$$\begin{aligned} |\lambda_j - \mathrm{e}^{-is}||\lambda_j - \mathrm{e}^{is}| &= |\mathrm{e}^{is_j} - \mathrm{e}^{-is}||\mathrm{e}^{is_j} - \mathrm{e}^{is}| \\ &= 4 \left| \sin \frac{s_j - s}{2} \sin \frac{s_j + s}{2} \right| = 2|\cos s - \cos s_j|, \end{aligned}$$

so that

$$I(s) = 4 \sum_{j=1}^{N} w_j^2 (\cos s - \cos s_j)^2 |q_{n-2}(\lambda_j)|^2$$

and therefore

$$I'(s_0) = -8 \sin s_0 \sum_{j=1}^{N} w_j^2 (\cos s_0 - \cos s_j) |q_{n-2}(\lambda_j)|^2. \tag{3.11}$$

Now let us assume that there are no eigenvalues on the open arc between θ_1 and θ_2. Then $0 < s_0 \leqslant |s_j| \leqslant \pi$ and so $\cos s_0 - \cos s_j \geqslant 0$ for every $j = 1, 2, \ldots, N$. There are at least $N - 2$ values of j with $0 < s_0 < |s_j| \leqslant \pi$ so that $\cos s_0 - \cos s_j > 0$ for at least one j. (We suppose $N > 2$ since otherwise $n \leqslant N - 1 \leqslant 1$ and there is nothing to prove.) It follows that all terms in the sum on the right-hand side of (3.11) are nonnegative and at least one is positive. Hence $I'(s_0) \neq 0$ (note $\sin s_0 \neq 0$, since $s_0 \in (0, \pi)$), which contradicts the fact that $I(s)$ has a minimum for $s = s_0$. The proposition is proved. ∎

Remark 3.5. Let θ_1 and θ_2 be as in the proof of Proposition 3.4. Let us denote the circular arc from θ_1 to θ_2 by $[\theta_1, \theta_2]$ and the complementary arc from θ_2 to θ_1 by $[\theta_2, \theta_1]$. Note that $\lambda_j \in [\theta_1, \theta_2]$ implies that $\cos s_0 - \cos s_j \leqslant 0$, so the fact that $I'(s_0) = 0$, where $I'(s_0)$ is given by (3.11), means that

$$\sum_{\lambda_j \in [\theta_1, \theta_2]} w_j^2 |\lambda_j - \theta_1||\lambda_j - \theta_2||q_{n-2}(\lambda_j)|^2 = \sum_{\lambda_j \in [\theta_2, \theta_1]} w_j^2 |\lambda_j - \theta_1||\lambda_j - \theta_2||q_{n-2}(\lambda_j)|^2.$$

We rewrite this in terms of the para-orthogonal polynomial ψ_n as

$$\sum_{\lambda_j \in [\theta_1, \theta_2]} w_j^2 \frac{|\psi_n(\lambda_j)|^2}{|\lambda_j - \theta_1||\lambda_j - \theta_2|} = \sum_{\lambda_j \in [\theta_2, \theta_1]} w_j^2 \frac{|\psi_n(\lambda_j)|^2}{|\lambda_j - \theta_1||\lambda_j - \theta_2|}. \tag{3.12}$$

There is an exact balance between the contributions from both arcs.

Remark 3.6. By now it is clear that the structure of unitary Hessenberg matrices with positive subdiagonal elements (connected to the IAP) is very similar to the structure of Jacobi matrices (connected to the Lanczos process). We have para-orthogonal polynomials instead of orthogonal polynomials, but both kinds of polynomials are characterized by a minimization problem and for both there is a separation property for their zeros. Since these properties of orthogonal polynomials were among the main tools in the study of the convergence of the Lanczos process in [18], we can use similar ideas for the convergence of the IAP, as will be clear from the proofs of the theorems that we give in the next section.

4. Proofs of Theorems 2.2, 2.3, and 2.4. Here we give the proofs of our main Theorems 2.2, 2.3, and 2.4. We will also make essential use of properties of logarithmic potentials U^μ. We refer the reader to [22, 23] for background information on logarithmic potential theory.

In what follows we use χ_p to denote the normalized zero counting measure of a polynomial p. So if p has degree n, then

$$\chi_p = \frac{1}{n} \sum_{p(\lambda)=0} \delta_\lambda,$$

where the sum is over all zeros of p and the zeros are counted according to their multiplicity.

Note that in section 3 we dropped the index N. Here it will reappear and we will use the properties and results of section 3 with no further comment.

4.1. Proof of Theorem 2.2. Theorem 2.2 was established for orthogonal polynomials whose zeros are on the real line by Rakhmanov [21]. Dragnev and Saff [11] used similar ideas to prove a more general theorem (including external fields), and weakened one of the conditions of Rakhmanov. Although these papers do not mention matrix iterations, we can nicely fit our setting in their results. The proof follows along arguments given in [11, 21]. We will indicate how we can modify them to the case of para-orthogonal polynomials, who have their zeros on the unit circle.

Proof of Theorem 2.2. Rakhmanov [21] showed that there exists a unique Borel probability measure μ_t that minimizes the logarithmic energy (2.9) among all Borel probability measures μ satisfying $0 \leqslant t\mu \leqslant \sigma$. He also showed that there exists a constant F_t such that (2.10) is satisfied and that (2.8) and (2.10) characterize the pair (μ_t, F_t). So we still need to prove (2.6) and (2.7).

The first step is to show that

$$\limsup_{\substack{n,N\to\infty \\ n/N\to t}} \|\psi_{n,N}(U_N)b_N\|^{1/n} \leqslant \mathrm{e}^{-F_t}. \tag{4.1}$$

The proof of (4.1) follows the proof of Lemma 5.3 in [11]. For a given $\varepsilon > 0$, a monic polynomial q_N of degree n with all its zeros on the unit circle is constructed for every large enough N so that

$$\limsup_{\substack{n,N\to\infty \\ n/N\to t}} \|q_N(U_N)b_N\|^{1/n} \leqslant \mathrm{e}^{-F_t+\varepsilon}. \tag{4.2}$$

There is a set $A \subset \mathbb{T}$ so that every eigenvalue of U_N outside A is a zero of q_N, and the rest of the zeros of q_N are taken in such a way that $\chi_{q_N} \to \mu_t$. We need to modify this construction slightly in order to guarantee that

$$q_N(0) = \psi_{n,N}(0) = \rho_{n,N}. \tag{4.3}$$

Since all the zeros of q_N are on the unit circle, $q_N(0)$ has unit modulus, and so we can achieve (4.3) by moving one of the zeros in A to a different position on the unit circle. This will not affect the estimate (4.2). Having (4.2) and (4.3) we use Theorem 3.3 to conclude that

$$\limsup_{\substack{n,N\to\infty \\ n/N\to t}} \|\psi_{n,N}(U_N)b_N\|^{1/n} \leqslant \mathrm{e}^{-F_t+\varepsilon}.$$

Since ε can be chosen arbitrarily small, (4.1) follows.

In the second step we establish the following. Suppose we are given a sequence $(q_N)_N$ of monic polynomials such that q_N has degree n, the zeros of q_N are separated by the eigenvalues of U_N and the normalized zero counting measures χ_{q_N} have a weak*-limit μ. Then

$$\liminf_{\substack{n,N\to\infty \\ n/N\to t}} \|q_N(U_N)b_N\|^{1/n} \geqslant \mathrm{e}^{-F^\mu}, \tag{4.4}$$

where

$$F^\mu = \min_{z\in\mathrm{supp}(\sigma-t\mu)} U^\mu(z). \tag{4.5}$$

Dragnev and Saff [11, Lemma 5.5] showed this for the case of the real line. The same proof works here.

In the third step we show that

$$F^\mu \leqslant F_t \tag{4.6}$$

for every Borel probability measure μ with $0 \leqslant t\mu \leqslant \sigma$, and equality in (4.6) holds if and only if $\mu = \mu_t$. Thus let μ be a Borel probability measure such that $0 \leqslant t\mu \leqslant \sigma$. Let $z \in \operatorname{supp}(\sigma - t\mu)$. We know from (2.10) that $U^{\mu_t}(z) \leqslant F_t$ and from (4.5) that $F^\mu \leqslant U^\mu(z)$. Hence

$$U^{\sigma - t\mu}(z) - U^{\sigma - t\mu_t}(z) = t(U^{\mu_t}(z) - U^\mu(z)) \leqslant t(F_t - F^\mu). \tag{4.7}$$

On $\mathbb{C} \setminus \operatorname{supp}(\sigma - t\mu)$ we have that $U^{\sigma - t\mu}$ is harmonic and $U^{\sigma - t\mu_t}$ superharmonic, so that $U^{\sigma - t\mu} - U^{\sigma - t\mu_t}$ is a subharmonic function there. Since $U^{\sigma - t\mu} - U^{\sigma - t\mu_t}$ is bounded at infinity (it has limit 0 at infinity), we can apply the maximum principle for subharmonic functions [22, Theorem 2.3.1], [23, Theorem 0.5.2], and it follows that (4.7) holds for every $z \in \mathbb{C}$. At infinity the left-hand side is 0, so that $F^\mu \leqslant F_t$.

If $F^\mu = F_t$, then we get $U^{\mu_t} - U^\mu \leqslant 0$ everywhere. Since at infinity these two functions are equal and their difference is a harmonic function on $\mathbb{C} \setminus \mathbb{T}$, we can conclude that it is zero outside the unit disc. By continuity, it is also zero on the unit circle. Inside the unit disc it is harmonic, and applying the maximum principle again, we find that it is zero inside the unit disc. So $U^{\mu_t} = U^\mu$ everywhere, which means that $\mu_t = \mu$ [22, Corollary 3.7.5], [23, Corollary II.2.2].

Now, collecting all the pieces finishes the proof. By Proposition 3.4, we know that the zeros of $\psi_{n,N}$ are separated by the eigenvalues of U_N. Let μ be a weak*-limit of a subsequence of the sequence of normalized zero counting measures $(\chi_{\psi_{n,N}})$. Then we find by (4.1) and (4.4) that

$$\mathrm{e}^{-F^\mu} \leqslant \liminf_{\substack{n,N\to\infty \\ n/N\to t}} \|\psi_{n,N}(U_N)b_N\|^{1/n} \leqslant \limsup_{\substack{n,N\to\infty \\ n/N\to t}} \|\psi_{n,N}(U_N)b_N\|^{1/n} \leqslant \mathrm{e}^{-F_t}; \tag{4.8}$$

hence $F^\mu \geqslant F_t$. From the separation property of the zeros of $\psi_{n,N}$ it also follows that $0 \leqslant t\mu \leqslant \sigma$. By (4.6) we must have $F^\mu = F_t$ so that $\mu = \mu_t$. Hence the inequalities in (4.8) are all equalities, which proves (2.7). We also see that μ_t is the only possible limit of a weak*-convergent subsequence of $(\chi_{\psi_{n,N}})$. Since the unit circle is compact, the set of Borel probability measures on $\mathbb{T}$ is compact in the weak*-topology. Hence the full sequence $(\chi_{\psi_{n,N}})$ converges to μ_t, which gives (2.6).

This concludes the proof of Theorem 2.2. ∎

4.2. Three lemmas. For the proof of Theorems 2.3 and 2.4 we need a number of lemmas. We will use the approach of Beckermann [5], who established these theorems for the Lanczos process. We will assume that the Conditions 2.1 hold.

The first lemma is borrowed from [6].

LEMMA 4.1 (see [6]). *Let σ be a Borel probability measure on the unit circle and suppose $(\Lambda_N)_N$ is a sequence of sets, all contained in $\mathbb{T}$, such that*

$$\lim_{N\to\infty} \frac{1}{N} \sum_{\lambda\in\Lambda_N} f(\lambda) = \int f(\lambda)\,\mathrm{d}\sigma(\lambda)$$

for every continuous function f on $\mathbb{T}$.

Let $t \in (0,1)$ and let μ be a Borel probability measure such that $t\mu \leqslant \sigma$. Let $n = n_N \leqslant \sharp\Lambda_N$ such that $n/N \to t$. Then there exists a sequence of sets $(Z_N)_N$ such that

(a) $\sharp Z_N = n$,

(b) $Z_N \subset \Lambda_N$, *and*

(c) *for all continuous functions f,*

$$\lim_{\substack{n,N\to\infty \\ n/N\to t}} \frac{1}{n} \sum_{\lambda\in Z_N} f(\lambda) = \int f(\lambda)\,\mathrm{d}\mu(\lambda).$$

Furthermore, if K is a closed set such that $\sigma(\partial K) = 0$ and $\sigma(K) = t\mu(K)$, then the sets Z_N can be chosen such that in addition to (a), (b), *and* (c), *we also have for N large enough,*

(d) $\Lambda_N \cap K \subset Z_N$.

Proof. In [6, Lemma A.1] this lemma is proven for the case where the sets Λ_N are contained in the real line. The same proof works here. ∎

The following lemma tells us that the eigenvalues inside $\Lambda(t,\sigma)$ are not exponentially close.

LEMMA 4.2 (see [5]). *We have*

$$\lim_{N\to\infty} \min\{|\lambda_{k\pm1,N} - \lambda_{k,N}|^{1/N} : k = 1,2,\dots,N\} = 1,$$

where we take $\lambda_{0,N} = \lambda_{N,N}$ and $\lambda_{N+1,n} = \lambda_{1,N}$.

Proof. In [5, Lemma 2.4(b)] this lemma is proven for the case of points on the real line. The same proof works here. ∎

The next lemma gives an estimate for $|\lambda_{k,N} - \theta_{\kappa-1,n,N}||\lambda_{k,N} - \theta_{\kappa,n,N}|$, where $\lambda_{k,N}$ is on the closed arc between $\theta_{\kappa-1,n,N}$ and $\theta_{\kappa,n,N}$. Recall that the isometric Ritz values are numbered counterclockwise and that $\theta_{0,n,N} := \theta_{n,n,N}$. We introduce the function

$$r_{\kappa,n,N}(z) = (z^{-1} - \bar\theta_{\kappa-1,n,N})(z - \theta_{\kappa,n,N})\sqrt{\theta_{\kappa-1,n,N}/\theta_{\kappa,n,N}}, \qquad \kappa = 1,\dots,n, \tag{4.9}$$

where we choose the branch of the square root belonging to the lower half plane. Thus, if $\theta_{\kappa-1,n,N} = \mathrm{e}^{i\tau_1}$ and $\theta_{\kappa,n,N} = \mathrm{e}^{i\tau_2}$ with $0 < \tau_2 - \tau_1 < 2\pi$, then

$$\sqrt{\theta_{\kappa-1,n,N}/\theta_{\kappa,n,N}} = \mathrm{e}^{-i\frac{\tau_2-\tau_1}{2}}. \tag{4.10}$$

Observe that $|\lambda_{k,N} - \theta_{\kappa-1,n,N}||\lambda_{k,N} - \theta_{\kappa,n,N}| = |r_{\kappa,n,N}(\lambda_{k,N})|$.

LEMMA 4.3. *Let $r_{\kappa,n,N}(z)$ be defined as in* (4.9)–(4.10). *Then the following hold.*

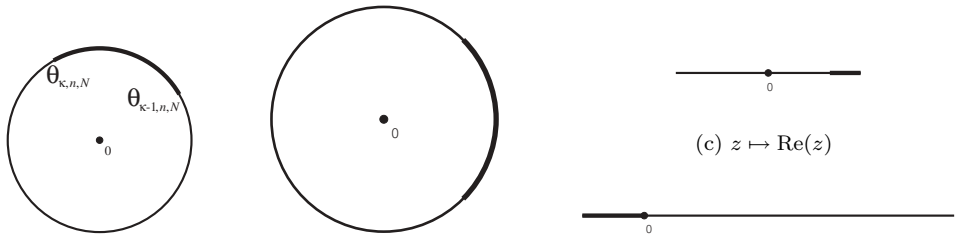
(a) *The function $r_{\kappa,n,N}(z)$ is real and negative for z on the open arc from $\theta_{\kappa-1,n,N}$ to $\theta_{\kappa,n,N}$ and real and positive on the complementary open arc.*

(b) *Let $\lambda_{k,N}$ be on the closed arc from $\theta_{\kappa-1,n,N}$ to $\theta_{\kappa,n,N}$. Then for every polynomial q of degree at most $n-2$,*

$$w_{k,N}^2|q(\lambda_{k,N})|^2|r_{\kappa,n,N}(\lambda_{k,N})| \leqslant \sum_{j\neq k} w_{j,N}^2|q(\lambda_{j,N})|^2 r_{\kappa,n,N}(\lambda_{j,N}). \tag{4.11}$$

(c) *Equality holds in* (4.11) *for the polynomial*

$$q(z) = \frac{\psi_{n,N}(z)}{(z-\theta_{\kappa-1,n,N})(z-\theta_{\kappa,n,N})}, \tag{4.12}$$



(a) The unit circle

(b) $z \mapsto \mathrm{e}^{-i\frac{\tau_1+\tau_2}{2}} z$

(d) $z \mapsto -2z + 2\cos\frac{\tau_2-\tau_1}{2}$

FIG. 4.1. *The image of the unit circle under the mapping $z \mapsto r_{\kappa,n,N}(z)$ step by step. Note that the arc between $\theta_{\kappa-1,n,N}$ and $\theta_{\kappa,n,N}$ is mapped to (part of) the negative real axis and the complementary arc to (part of) the positive real axis.*

where $\psi_{n,N}$ is the monic para-orthogonal polynomial.

Proof. Let $z \in \mathbb{T}$ and choose $\tau_1 = \arg\theta_{\kappa-1,n,N}$, $\tau_2 = \arg\theta_{\kappa,n,N}$ such that $0 < \tau_2 - \tau_1 < 2\pi$. Then we have for $z \in \mathbb{T}$,

$$\begin{aligned} r_{\kappa,n,N}(z) &= (z - \mathrm{e}^{i\tau_2})(\bar{z} - \mathrm{e}^{-i\tau_1})\mathrm{e}^{-i\frac{\tau_2-\tau_1}{2}} \\ &= (\mathrm{e}^{-i\frac{\tau_1+\tau_2}{2}} z - \mathrm{e}^{i\frac{\tau_2-\tau_1}{2}})(\mathrm{e}^{i\frac{\tau_1+\tau_2}{2}} \bar{z} - \mathrm{e}^{i\frac{\tau_2-\tau_1}{2}})\mathrm{e}^{-i\frac{\tau_2-\tau_1}{2}} \\ &= \mathrm{e}^{-i\frac{\tau_2-\tau_1}{2}} - \mathrm{e}^{-i\frac{\tau_1+\tau_2}{2}} z - \mathrm{e}^{i\frac{\tau_1+\tau_2}{2}} \bar{z} + \mathrm{e}^{i\frac{\tau_2-\tau_1}{2}} \\ &= -2\,\mathrm{Re}(\mathrm{e}^{-i\frac{\tau_1+\tau_2}{2}} z) + 2\cos\frac{\tau_2-\tau_1}{2}. \end{aligned}$$

This shows that $r_{\kappa,n,N}(z)$ is real for $z \in \mathbb{T}$. Moreover, $r_{\kappa,n,N}$ is the composition of the mappings $z \mapsto e^{-i\frac{\tau_1+\tau_2}{2}} z$, $z \mapsto \mathrm{Re}\, z$, and $z \mapsto -2z + 2\cos\frac{\tau_2-\tau_1}{2}$. The effect of these mappings on the unit circle is plotted step by step in Figure 4.1. Following these mappings, we obtain the statements of part (a).

To prove part (b), we use the Gaussian quadrature formula (3.9). We know that there exist positive real numbers $\beta_{1,N}, \ldots, \beta_{n,N}$ such that

$$\sum_{j=1}^{N} w_{j,N}^2 p(\lambda_{j,N}) = \sum_{j=1}^{n} \beta_{j,N} p(\theta_{j,n,N}) \tag{4.13}$$

for every Laurent polynomial p of degree $n-1$. Now let q be a polynomial of degree at most $n-2$ and write

$$p(z) = r_{\kappa,n,N}(z) q(z) \bar{q}(z^{-1}), \tag{4.14}$$

where $\bar{q}$ is the polynomial whose coefficients are the complex conjugates of the coefficients of q. This p is a Laurent polynomial of degree $n-1$, so we can apply (4.13) to p. Because of part (a), we know that $r_{\kappa,n,N}(\theta_{j,n,N}) \geqslant 0$ for all j. Since also $q(z)\bar{q}(z^{-1}) = |q(z)|^2 \geqslant 0$ for all $z \in \mathbb{T}$, we see that $p(\theta_{j,n,N}) \geqslant 0$ for all j. So the right-hand side of (4.13) is nonnegative, which implies that

$$-w_{k,N}^2 p(\lambda_{k,N}) \leqslant \sum_{j \neq k} w_{j,N}^2 p(\lambda_{j,N}), \tag{4.15}$$

which gives

$$-w_{k,N}^2 r_{\kappa,n,N}(\lambda_{k,N})|q(\lambda_{k,N})|^2 \leqslant \sum_{j\neq k} w_{j,N}^2 r_{\kappa,n,N}(\lambda_{j,N})|q(\lambda_{j,N})|^2. \tag{4.16}$$

Now $r_{\kappa,n,N}(\lambda_{k,N}) < 0$ according to part (a) again, since $\lambda_{k,N}$ is on the arc from $\theta_{\kappa-1,n,N}$ to $\theta_{\kappa,n,N}$. Using this in (4.16) we obtain (4.11). This proves part (b).

Finally, if we use the polynomial q from (4.12) in the construction (4.14), then the right-hand side of (4.13) equals zero, since all terms vanish. This leads to equality in (4.11), so that part (c) follows. ∎

For every polynomial q of degree at most $n-2$ with $q(\lambda_{k,N}) \neq 0$, (4.11) can be rewritten as

$$\begin{aligned}|\lambda_{k,N} - \theta_{\kappa-1,n,N}||\lambda_{k,N} - \theta_{\kappa,n,N}| \\ \leqslant \frac{\sum_{j\neq k} w_{j,N}^2(\bar\lambda_{j,N} - \bar\theta_{\kappa-1,n,N})(\lambda_{j,N} - \theta_{\kappa,n,N})\sqrt{\theta_{\kappa-1,n,N}/\theta_{\kappa,n,N}}|q(\lambda_{j,N})|^2}{w_{k,N}^2|q(\lambda_{k,N})|^2}.\end{aligned} \tag{4.17}$$

From this we deduce

$$\begin{aligned}\min_j|\lambda_{k,N} - \theta_{j,n,N}| &\leqslant \big(|\lambda_{k,N} - \theta_{\kappa-1,n,N}||\lambda_{k,N} - \theta_{\kappa,n,N}|\big)^{1/2} \\ &\leqslant \left(\frac{\sum_{j\neq k} w_{j,N}^2|\bar\lambda_{j,N} - \bar\theta_{\kappa-1,n,N}||\lambda_{j,N} - \theta_{\kappa,n,N}||q(\lambda_{j,N})|^2}{w_{k,N}^2|q(\lambda_{k,N})|^2}\right)^{1/2} \\ &\leqslant \left(\frac{\max_{j\neq k}|q(\lambda_{j,N})|}{|q(\lambda_{k,N})|}\right)\left(4\frac{\sum_{j\neq k} w_{j,N}^2}{w_{k,N}^2}\right)^{1/2}.\end{aligned} \tag{4.18}$$

4.3. Proof of Theorem 2.3. To prove Theorem 2.3, we use the estimate (4.18). We are going to find estimates for the numerator and denominator of the first factor in the right-hand side. To this end we will construct a suitable polynomial q.

Proof of Theorem 2.3. Let $(k_N)_N$ be a sequence of indices so that $\lim_{N\to\infty}\lambda_{k_N} = \lambda$. Since all eigenvalues and all isometric Ritz values are contained in the unit circle, there is nothing to prove if $U^{\mu_t}(\lambda) = F_t$.

So suppose $U^{\mu_t}(\lambda) < F_t$ and let $\varepsilon \in (0, -U^{\mu_t}(\lambda) + F_t)$. Define

$$K := \{z \in \mathbb{T} \mid -U^{\mu_t}(z) + F_t \geqslant \varepsilon\}.$$

Since U^σ is continuous, so is U^{μ_t} (see, e.g., [11, Lemma 5.2]), so that K is closed and contains an η-neighborhood of λ (we take $\eta < 1$). Now $K \cap \operatorname{supp}(\sigma - t\mu_t) = \emptyset$, so $\sigma(K) = t\mu_t(K)$. Without loss of generality we may suppose that $\sigma(\partial K) = 0$ (see also Remark 4.4 below). We can now obtain a sequence of sets $(Z_N)_N$ by Lemma 4.1 with $\mu = \mu_t$ and n replaced by $n-1$.

By Condition 2.1(2) we can choose $\delta < \eta$ such that (2.2) holds for N sufficiently large and for all $k \leqslant N$. Note that by properties (b) and (d) of Lemma 4.1 and the definition of K, all eigenvalues $\lambda_{j,N}$ with $|\lambda_{j,N} - \lambda_{k_N,N}| < \delta$ are in Z_N, when N is large enough. We define

$$q_N(z) := \prod_{\lambda_{j,N}\in Z_N'} (z - \lambda_{j,N}),$$

where $Z'_N := Z_N \setminus \{\lambda_{k_N,N}\}$ (so q_N is a polynomial of degree $n-2$). Note that property (c) of Lemma 4.1 still holds when we replace the sets Z_N by Z'_N, i.e., the sequence of normalized zero counting measures of $(q_N)_N$ converges in weak*-sense to μ_t.

We factor q_N in two parts, one containing the zeros close to $\lambda_{k_N,N}$ and one containing the other zeros:

$$q_N^{(1)}(z) := \prod_{0<|\lambda_{j,N}-\lambda_{k_N,N}|<\delta} (z-\lambda_{j,N}), \qquad q_N^{(2)}(z) := \frac{q_N(z)}{q_N^{(1)}(z)}.$$

We also define the measures

$$\mu_N^{(1)} := \frac{1}{n-2} \sum_{q_N^{(1)}(\lambda)=0} \delta_\lambda, \qquad \mu_N^{(2)} := \frac{1}{n-2} \sum_{q_N^{(2)}(\lambda)=0} \delta_\lambda.$$

Then $\chi_{q_N} = \mu_N^{(1)} + \mu_N^{(2)}$, so that

$$U^{\chi_{q_N}}(\lambda_{k_N,N}) = U^{\mu_N^{(1)}}(\lambda_{k_N,N}) + U^{\mu_N^{(2)}}(\lambda_{k_N,N}). \tag{4.19}$$

Because of Condition 2.1(2),

$$U^{\mu_N^{(1)}}(\lambda_{k_N,N}) < \varepsilon \tag{4.20}$$

for N large enough. Since $\lim_{n,N\to\infty,n/N\to t} \mu_N^{(2)} = \mu_t|_{\mathbb{T}\setminus B(\lambda,\delta)}$, we get

$$\lim_{\substack{n,N\to\infty \\ n/N\to t}} U^{\mu_N^{(2)}}(\lambda_{k_N,N}) = U^{\mu_t|_{\mathbb{T}\setminus B(\lambda,\delta)}}(\lambda) = U^{\mu_t}(\lambda) - U^{\mu_t|_{B(\lambda,\delta)}}(\lambda) \leqslant U^{\mu_t}(\lambda), \tag{4.21}$$

where the last inequality holds since $\delta < 1$. Combining the two estimates (4.20) and (4.21) with (4.19), we get

$$\limsup_{\substack{n,N\to\infty \\ n/N\to t}} U^{\chi_{q_N}}(\lambda_{k_N,N}) \leqslant U^{\mu_t}(\lambda) + \varepsilon, \tag{4.22}$$

so that

$$\liminf_{\substack{n,N\to\infty \\ n/N\to t}} \log|q_N(\lambda_{k_N,N})|^{1/n} = \liminf_{\substack{n,N\to\infty \\ n/N\to t}} -U^{\chi_{q_N}}(\lambda_{k_N,N}) \geqslant -U^{\mu_t}(\lambda) - \varepsilon. \tag{4.23}$$

Now we are going to estimate the absolute value of q_N on the rest of the spectrum of U_N. By construction, $q_N(\lambda_{j,N}) = 0$ for $\lambda_{j,N} \in K \setminus \{\lambda_{k_N,N}\}$, so we have

$$\max_{j\neq k_N} |q_N(\lambda_{j,N})| = \max_{\lambda_{j,N}\notin K} |q_N(\lambda_{j,N})| \leqslant \sup_{z\in\mathbb{T}\setminus K} |q_N(z)|.$$

Since the zero distributions of q_N converge to μ_t, we can apply the principle of descent [23, Theorem I.6.8]. Then we get

$$\begin{aligned} \limsup_{\substack{n,N\to\infty \\ n/N\to t}} \max_{j\neq k_N} \log|q_N(\lambda_{j,N})|^{1/n} &\leqslant \limsup_{\substack{n,N\to\infty \\ n/N\to t}} \sup_{z\in\mathbb{T}\setminus K} \frac{1}{n} \log|q_N(z)| \\ &\leqslant \sup_{z\in\mathbb{T}\setminus K} \big(-U^{\mu_t}(z)\big) \leqslant -F_t + \varepsilon, \end{aligned} \tag{4.24}$$

where the last inequality follows from the definition of K.

If we now choose $q = q_N$ and $k = k_N$ in (4.18), we get

$$\limsup_{\substack{n,N\to\infty \\ n/N\to t}} \min_j |\lambda_{k_N,N} - \theta_{j,n,N}|^{1/n} \leqslant \limsup_{\substack{n,N\to\infty \\ n/N\to t}} \left[\left(\frac{\max_{j\neq k_N} |q_N(\lambda_{j,N})|}{|q_N(\lambda_{k_N,N})|} \right)^{1/n} \left(4 \frac{\sum_{j\neq k_N} w_{j,N}^2}{w_{k_N,N}^2} \right)^{1/2n} \right], \tag{4.25}$$

The second factor in the lim sup on the right-hand side of (4.25) converges to 1, because of Condition 2.1(3), while the first factor is handled by (4.23) and (4.24). The result is that

$$\limsup_{\substack{n,N\to\infty \\ n/N\to t}} \min_j |\lambda_{k_N,N} - \theta_{j,n,N}|^{1/n} \leqslant \exp\bigl(U^{\mu_t}(\lambda) - F_t + 2\varepsilon\bigr).$$

Since this holds for all $\varepsilon > 0$, (2.11) is proven. □

Remark 4.4. If the set K is Cantor-like and the measure σ singular, we might have that $\sigma(\partial K) > 0$. However, since

$$\partial K \subseteq \{z \in \mathbb{T} \mid -U^{\mu_t}(z) + F_t = \varepsilon\}$$

we have that $\sigma(\partial K) > 0$ can only happen for a countable number of ε's. So if $\sigma(\partial K) > 0$, we can choose a smaller ε so that $\sigma(\partial K) = 0$ and continue with the proof of Theorem 2.3.

4.4. Proof of Theorem 2.4. We finally give the proof of Theorem 2.4. As noted before, the proof is based on the proof of [5, Theorem 2.1], but we have streamlined some of the arguments.

Proof of Theorem 2.4. Since $\lambda \in \Lambda(t,\sigma)$ we have $F_t - U^{\mu_t}(\lambda) > 0$. By continuity there is a δ-neighborhood Δ_δ of λ and an $\varepsilon > 0$ such that $F_t - U^{\mu_t}(z) > \varepsilon$ for $z \in \Delta_\delta$. Because of Theorem 2.3 we then know that each eigenvalue $\lambda_{k,N} \in \Delta_\delta$ has an isometric Ritz value close to it if N is large. More precisely, we can ensure that

$$\min_j |\lambda_{k,N} - \theta_{j,n,N}| \leqslant \mathrm{e}^{-n\varepsilon}$$

for all $\lambda_{k,N} \in \Delta_\delta$ if N is large enough.

Now we study the relative positions of eigenvalues and isometric Ritz values. Using (i) the separation property (see Proposition 3.4), (ii) the fact that eigenvalues are exponentially well approximated (see Theorem 2.3), and (iii) the fact that the distance between eigenvalues is *not* exponentially small (see Lemma 4.2), we can make a complete classification of these relative positions for N large enough. The different cases were plotted in Figure 2.1. The exceptions are covered below and illustrated in Figure 4.2.

From the separation property we conclude that close to an eigenvalue there can be at most two isometric Ritz values (one on either side of it on the unit circle). However, it is easily seen that at most one eigenvalue $\lambda_{\ell_1,N}$ in Δ_δ can be approximated by two isometric Ritz values, again because the isometric Ritz values are separated by the eigenvalues and because each eigenvalue is well approximated by at least one isometric Ritz value. In this case we define the exceptional index as $k_N^*(\lambda) := \ell_1$.

(a) Case 2: The exceptional eigenvalue is approximated by two isometric Ritz values.

(b) Case 5: The exceptional eigenvalue coincides with an isometric Ritz value.

(c) Case 6: The exceptional eigenvalue is one of the two eigenvalues indicated; see (4.26).

FIG. 4.2. *The definition of the exceptional indices in the different cases (see Figure* 2.1*). In Cases* 1, 3, *and* 4 *no exceptions need to be made.*

Another possibility is that an eigenvalue $\lambda_{\ell_2,N}$ and an isometric Ritz value coincide. In similar fashion one can see that this happens at most once, and that this case is not compatible with the previous one. Then we define the exceptional index as $k_N^*(\lambda) := \ell_2$.

It is also possible that there are two consecutive eigenvalues, $\lambda_{\ell_3-1,N}$ and $\lambda_{\ell_3,N}$, in Δ_δ that do not have an isometric Ritz value on the arc between them. Again, it is easily seen that this can happen only once in Δ_δ and that this excludes the previous two possibilities. In this case the exceptional index is either $\ell_3 - 1$ or ℓ_3, depending on the proximity of the nearest isometric Ritz value. More precisely, let $\theta_{\kappa,n,N}$ be the first isometric Ritz value after $\lambda_{\ell_3,N}$. We define the exceptional index as

$$k_N^*(\lambda) := \begin{cases} \ell_3 & \text{if } |\theta_{\kappa,n,N} - \lambda_{\ell_3,N}| \leqslant |\theta_{\kappa-1,n,N} - \lambda_{\ell_3-1,N}|, \\ \ell_3 - 1 & \text{otherwise.} \end{cases} \tag{4.26}$$

Now if $\lambda_{k_N,N} \to \lambda$, then for N large enough $\lambda_{k_N,N} \in \Delta_\delta$. Furthermore, if $k_N \neq k_N^*(\lambda)$, there is exactly one isometric Ritz value $\theta_{j,n,N}$ close to $\lambda_{k_N,N}$ (Case 2 is the only exception to this). All other isometric Ritz values are at a distance whose nth root limit is 1 (see Lemma 4.2). It then follows that (4.18) can be sharpened to

$$\min_j |\lambda_{k_N,N} - \theta_{j,n,N}| \leqslant c_{n,N} \left(\frac{\max_{j \neq k} |q(\lambda_{j,N})|}{|q(\lambda_{k,N})|} \right)^2,$$

with constants $c_{n,N}$ such that $\lim_{n,N\to\infty, n/N\to t} c_{n,N}^{1/n} = 1$. Examining the proof of Theorem 2.3, we see that this leads to

$$\limsup_{\substack{n,N\to\infty \\ n/N\to t}} \min_j |\lambda_{k_N,N} - \theta_{j,n,N}|^{1/n} \leqslant \exp\Big(2\big(U^{\mu_t}(\lambda) - F_t\big)\Big). \tag{4.27}$$

Next we prove the lower bound for $\min_j |\lambda_{k_N,N} - \theta_{j,n,N}|^{1/n}$ when $k_N \neq k_N^*(\lambda)$. Choose κ such that $\lambda_{k_N,N}$ is on the arc from $\theta_{\kappa-1,n,N}$ to $\theta_{\kappa,n,N}$. From Remark 3.5 it follows that

$$\sum_{\lambda_{j,N} \in [\theta_{\kappa-1,n,N}, \theta_{\kappa,n,N}]} w_{j,N}^2 \frac{|\psi_{n,N}(\lambda_{j,N})|^2}{|\lambda_{j,N} - \theta_{\kappa-1,n,N}||\lambda_{j,N} - \theta_{\kappa,n,N}|}$$
$$= \sum_{\lambda_{j,N} \notin [\theta_{\kappa-1,n,N}, \theta_{\kappa,n,N}]} w_{j,N}^2 \frac{|\psi_{n,N}(\lambda_{j,N})|^2}{|\lambda_{j,N} - \theta_{\kappa-1,n,N}||\lambda_{j,N} - \theta_{\kappa,n,N}|},$$

where $[\theta_{\kappa-1,n,N}, \theta_{\kappa,n,N}]$ denotes the circular arc going from $\theta_{\kappa-1,n,N}$ to $\theta_{\kappa,n,N}$. Thus

$$\sum_{\lambda_{j,N}\in[\theta_{\kappa-1,n,N},\theta_{\kappa,n,N}]} w_{j,N}^2 \frac{|\psi_{n,N}(\lambda_{j,N})|^2}{|\lambda_{j,N}-\theta_{\kappa-1,n,N}||\lambda_{j,N}-\theta_{\kappa,n,N}|}$$
$$= \frac{1}{2}\sum_{j=1}^{N} w_{j,N}^2 \frac{|\psi_{n,N}(\lambda_{j,N})|^2}{|\lambda_{j,N}-\theta_{\kappa-1,n,N}||\lambda_{j,N}-\theta_{\kappa,n,N}|}$$
$$\geqslant \frac{1}{8}\sum_{j=1}^{N} w_{j,N}^2 |\psi_{n,N}(\lambda_{j,N})|^2 = \frac{1}{8}\|\psi_{n,N}(U_N)b_N\|^2.$$

Because of the limit (2.7) in Theorem 2.2, it then follows that

(4.28)
$$\liminf_{\substack{n,N\to\infty\\ n/N\to t}} \left(\sum_{\lambda_{j,N}\in[\theta_{\kappa-1,n,N},\theta_{\kappa,n,N}]} w_{j,N}^2 \frac{|\psi_{n,N}(\lambda_{j,N})|^2}{|\lambda_{j,N}-\theta_{\kappa-1,n,N}||\lambda_{j,N}-\theta_{\kappa,n,N}|} \right)^{1/n} \geqslant \exp(-2F_t).$$

The sum on the left-hand side has at most two terms, one of them for $\lambda_{k_N,N}$.

If there is only one term in the sum on the left-hand side of (4.28) (Cases 1, 2, 3, and 4) or if one of the terms is 0 (Case 5), then (4.28) says

$$\liminf_{\substack{n,N\to\infty\\ n/N\to t}} \left(w_{k_N,N}^2 \frac{|\psi_{n,N}(\lambda_{k_N,N})|^2}{|\lambda_{k_N,N}-\theta_{\kappa-1,n,N}||\lambda_{k_N,N}-\theta_{\kappa,n,N}|} \right)^{1/n} \geqslant \exp(-2F_t). \tag{4.29}$$

Note that $\frac{\psi_{n,N}(z)}{(z-\theta_{\kappa-1,n,N})(z-\theta_{\kappa,n,N})}$ is a monic polynomial of degree $n-2$ with roots $\theta_{j,n,N}$, $j \neq \kappa-1, \kappa$. From (2.6) it follows that μ_t is the weak*-limit of the normalized zero counting measures of these polynomials, and from this it follows that, by the principle of descent [23, Theorem I.6.8]),

$$\limsup_{\substack{n,N\to\infty\\ n/N\to t}} \left(\frac{|\psi_{n,N}(\lambda_{k_N,N})|}{|\lambda_{k_N,N}-\theta_{\kappa-1,n,N}||\lambda_{k_N,N}-\theta_{\kappa,n,N}|} \right)^{1/n} \leqslant \exp(-U^{\mu_t}(\lambda)). \tag{4.30}$$

Using $\lim_{\substack{n,N\to\infty\\ n/N\to t}} w_{k_N,N}^{1/n} = 1$, we obtain from (4.29) that

$$\liminf_{\substack{n,N\to\infty\\ n/N\to t}} (|\lambda_{k_N,N}-\theta_{\kappa-1,n,N}||\lambda_{k_N,N}-\theta_{\kappa,n,N}|)^{1/n}$$
$$\geqslant \exp(-2F_t) \liminf_{\substack{n,N\to\infty\\ n/N\to t}} \left(\frac{|\lambda_{k_N,N}-\theta_{\kappa-1,n,N}||\lambda_{k_N,N}-\theta_{\kappa,n,N}|}{|\psi_{n,N}(\lambda_{k_N,N})|} \right)^{2/n},$$

and together with (4.30) this gives us

$$\liminf_{\substack{n,N\to\infty\\ n/N\to t}} (|\lambda_{k_N,N}-\theta_{\kappa-1,n,N}||\lambda_{k_N,N}-\theta_{\kappa,n,N}|)^{1/n} \geqslant \exp\Big(2\big(U^{\mu_t}(\lambda)-F_t\big)\Big). \tag{4.31}$$

Now we can conclude

$$\begin{aligned}(4.32)\quad \liminf_{\substack{n,N\to\infty\\ n/N\to t}} \min_j |\lambda_{k_N,N}-\theta_{j,n,N}|^{1/n} &\geqslant \liminf_{\substack{n,N\to\infty\\ n/N\to t}} \left(\frac{|\lambda_{k_N,N}-\theta_{\kappa-1,n,N}||\lambda_{k_N,N}-\theta_{\kappa,n,N}|}{2}\right)^{1/n}\\ &\geqslant \exp\Big(2\big(U^{\mu_t}(\lambda)-F_t\big)\Big).\end{aligned}$$

The other possibility is that there are two terms in the sum on the left-hand side of (4.28). Then we are in Case 6. Let k'_N be the index j giving the largest term in the sum. Then

$$\liminf_{\substack{n,N\to\infty\\ n/N\to t}} \left(w^2_{k'_N,N}\frac{|\psi_{n,N}(\lambda_{k'_N,N})|^2}{|\lambda_{k'_N,N}-\theta_{\kappa-1,n,N}||\lambda_{k'_N,N}-\theta_{\kappa,n,N}|}\right)^{1/n} \geqslant \exp(-2F_t)$$

and from this it follows as before that

$$(4.33)\qquad \liminf_{\substack{n,N\to\infty\\ n/N\to t}} \min_j |\lambda_{k'_N,N}-\theta_{j,n,N}|^{1/n} \geqslant \exp\Big(2\big(U^{\mu_t}(\lambda)-F_t\big)\Big).$$

Since $\min_j|\lambda_{k_N,N}-\theta_{j,n,N}| \geqslant \min_j|\lambda_{k'_N,N}-\theta_{j,n,N}|$ (by the definition of k^*_N in (4.26)), we also obtain (4.32) in this case.

Therefore we have (4.32) in both cases. Together with (4.27) this proves equation (2.12). ∎

5. Numerical experiments. For the numerical experiments, we take a large unitary matrix U of size $N\times N$ and we execute the IAP for every $n\leqslant N$ (so we let $t=n/N$ vary from 0 to 1).

Our theoretical results are independent of the choice of the parameters $\rho_{n,N}$, but for the experiments we have to make a choice. We choose $\rho_{n,N}=\gamma_{n,N}/|\gamma_{n,N}|$, as this choice assures the modified submatrix stays as close as possible to the original submatrix [16, Lemma 2.1].

The experiments were done on matrices U whose eigenvalues are distributed according to a combination of von Mises distributions. A von Mises distribution is a continuous distribution on $\mathbb{T}$ with density

$$P(e^{i\theta})=\frac{1}{2\pi I_0(\alpha)}\mathrm{e}^{\alpha\cos(\theta-\theta_0)},$$

where I_0 is the modified Bessel function of the first kind and order 0. We have that θ_0 is the *mean direction* and α is the *concentration parameter*. Von Mises distributions appear in directional statistics [20].

For the experiments we used MATLAB. Codes for unitary Hessenberg QR (UHQR) were kindly provided to us by William B. Gragg and Michael Stewart. Random numbers from the von Mises distributions were generated using the R environment.[1] We sorted a very large sample of size mN and selected every mth point from it. A typical value of m we used was $m=4000$. The points were then used as the eigenvalues

[1]The R project for statistical computing, http://www.r-project.org.

of an $N \times N$ unitary matrix to which we applied the IAP. We followed this procedure in order to obtain eigenvalues that follow the limiting distribution adequately. For the matrix sizes we used (namely, $N = 300$), a fully random sample does not follow the limiting distribution very well and our asymptotical results do not apply.

5.1. Distribution of isometric Ritz values. To improve the understanding of the experiments, we recall the minimizing property of μ_t; see Theorem 2.2. If we minimize $I(\mu)$ among *all* Borel probability measures supported on $\mathbb{T}$ then the solution is the normalized Lebesgue measure on $\mathbb{T}$ [23, p. 25], which we denote here by λ. Now if t is so small that $t\lambda < \sigma$, then μ_t is equal to λ, because of their respective minimizing properties. So then everywhere $t\mu_t < \sigma$, so that no convergence can be expected (see the discussion after Theorem 2.3). If t grows, $t\lambda$ will also increase, until for a certain critical t_{cr} it hits σ at the point (or points) where the density of σ is minimal. For slightly larger $t > t_{cr}$, eigenvalues will be found in a neighborhood of that point (or those points), since eigenvalues are found where $t\mu_t = \sigma$. If we let t increase further, the region of good convergence also increases.

Continuing this line of thought, one might think that convergence will be slowest in the region where the eigenvalue density has its maximum. However, this is not necessarily true (although in many cases it is), since μ_t might be very different from the normalized Lebesgue measure when t is not small.

In Figure 5.1 we present an example. The eigenvalues of the 300×300 matrix U are distributed according to a combination of three von Mises distributions, with respective parameter pairs

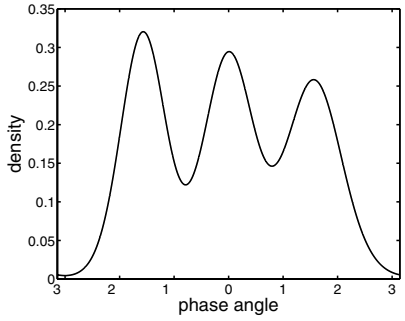
$$(\theta_0, \alpha) = (-\pi/2, 6), (0, 5) \text{ and } (\pi/2, 4).$$

The density is shown in part (a) of Figure 5.1. The distribution has three local maxima near the values $-i$, 0, and i, that is, the points with angles $-\pi/2$, 0, and $\pi/2$. Part (b) shows the convergence plot for the IAP. A + is plotted for every isometric Ritz value whose distance to its nearest eigenvalue is less than 10^{-5}.

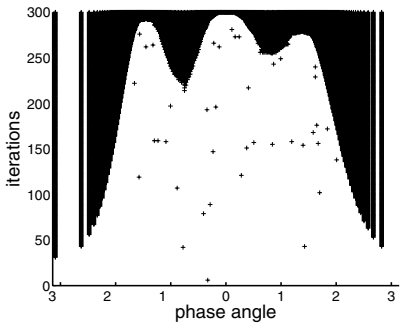
It can be seen that the shape of the convergence plot resembles the eigenvalue density. For the regions of low eigenvalue density, this follows from the preceding discussion. If we look at higher values of t (i.e., more iterations), then we see a difference between the two plots. The eigenvalue density has a maximum near $-\frac{\pi}{2}$, but the eigenvalues near that maximum are approximated earlier than eigenvalues near 0, where the peak is lower. To explain this phenomenon, we plotted a simulation of $t\mu_t$ for $t = 0.4$ in Figure 5.2. In the region where the constraint σ is active, μ_t does not look like λ at all, which is rather obvious (it is prohibited to do so by the constraint σ). Figure 5.2 shows that eigenvalues in the peaks around $-\pi/2$ and $\pi/2$ are indeed found earlier than eigenvalues in the peak around 0.

5.2. Convergence speed. Now we will check the assertions of Theorem 2.4. If we assume the right-hand side of (2.12) is constant as a function of t, we expect linear convergence. In fact, that right-hand side is slightly decreasing, so we should be able to observe a superlinear convergence (this superlinearity is of the same nature as the one discussed in [6, 8]). In Figure 5.3(a) the convergence graphs are plotted and indeed the (super)linearity appears.

We also tried to generate Case 2 from the classification made in Figure 2.1 also shown in Figure 4.2(a); in this case two isometric Ritz values are close to the same eigenvalue. Remember that this was an exception to the doubled convergence rate in Theorem 2.4. We created a real orthogonal matrix, with 1 as an eigenvalue. In the



(a) The distribution of eigenvalues



(b) Convergence of isometric Ritz values

FIG. 5.1. *Convergence result for the IAP applied on a* 300×300 *matrix* U *with eigenvalues distributed as in* (a). *In* (b) *the iteration step (dimension of the modified Hessenberg submatrix) is plotted on the* Y*-axis and the phase angle of the eigenvalues on the* X*-axis. If an isometric Ritz value is closer to an eigenvalue than* 10^{-5}, *a "+" is plotted.*

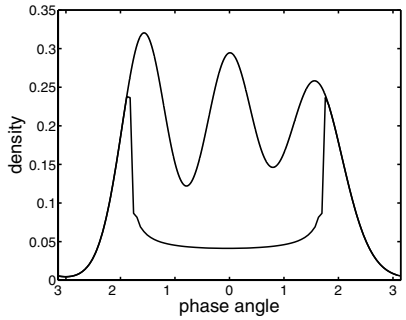
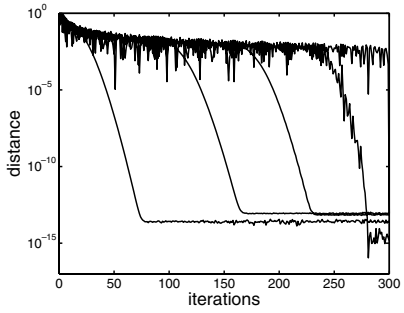
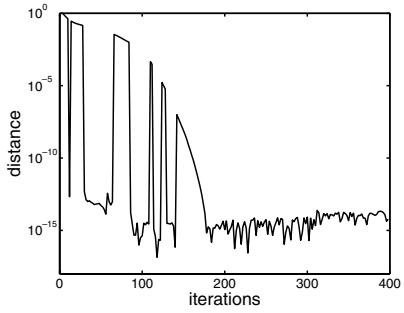


FIG. 5.2. *A simulation of* $t\mu_t$ *when* σ *is as in Figure* 5.1(a) *for* $t = 0.4$.



(a) Convergence graph of five eigenvalues of the example of Figure 5.1. The eigenvalues that are approximated first clearly show the superlinear behavior.



(b) Convergence graph for an example with the exception (2.13a) of Theorem 2.4. The distance to the eigenvalue 1 is either very small (if an isometric Ritz value is 1 up to machine precision) or relatively large (if a pair of complex conjugate isometric Ritz values approximates 1).

FIG. 5.3. *Convergence graphs of individual eigenvalues.*

isometric Arnoldi process we took $\rho_{n,N} = 1$ so that the Hessenberg matrices are real and nonreal isometric Ritz values come in conjugate pairs. So in each step there are two possibilities. Either there is an isometric Ritz value at 1, or there is a pair of complex conjugate isometric Ritz values closest to 1. In the latter case we have the exception (2.13a). The convergence is then slower as in the typical case.

In Figure 5.3(b) we see that both kinds of behavior do happen, depending on the value of n. If there is an isometric Ritz value at 1, then the distance is of course small (zero up to machine precision), while the distance is much bigger if there is a pair of complex conjugate Ritz values closest to 1. The graph only shows the results for an even number of iterations, because in this example one isometric Ritz value turned out to be 1 in every odd step starting at about $n = 25$.

Acknowledgments. We thank Bernhard Beckermann for allowing us to use the material from [5]. We thank William B. Gragg and Michael Stewart for providing us with codes for UHQR. We thank Bernhard Beckermann and Walter Van Assche for useful discussions.

REFERENCES

[1] G.S. Ammar and C. He, *On an inverse eigenvalue problem for unitary Hessenberg matrices*, Linear. Algebra Appl., 218 (1995), pp. 263–271.

[2] J. Baik, T. Kriecherbauer, K. T.-R. McLaughlin, and P. D. Miller, *Uniform asymptotics for polynomials orthogonal with respect to a general class of discrete weights and universality results for associated ensembles: Announcement of results*, Int. Math. Res. Not., 2003 (2003), pp. 821–858.

[3] J. Baik, T. Kriecherbauer, K. T.-R. McLaughlin, and P. D. Miller, *Uniform asymptotics for polynomials orthogonal with respect to a general class of discrete weights and universality results for associated ensembles*, preprint math.CA/0310278.

[4] C. Beattie, M. Embree, and J. Rossi, *Convergence of restarted Krylov subspaces to invariant subspaces*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 1074–1109.

[5] B. Beckermann, *A note on the convergence of Ritz values for sequences of matrices*, Publication ANO 408, Université de Lille, France, 2000.

[6] B. Beckermann and A. B. J. Kuijlaars, *Superlinear convergence of conjugate gradients*, SIAM J. Numer. Anal., 39 (2001), pp. 300–329.

[7] B. Beckermann and A. B. J. Kuijlaars, *On the sharpness of an asymptotic error estimate for conjugate gradients*, BIT, 41 (2001), pp. 856–867.

[8] B. Beckermann and A. B. J. Kuijlaars, *Superlinear CG convergence for special right-hand sides*, Electron. Trans. Numer. Anal., 14 (2002), pp. 1–19.

[9] A. Bunse-Gerstner and H. Fassbender, *Error bounds in the isometric Arnoldi process*, J. Comput. Appl. Math., 86 (1997), pp. 53–72.

[10] A. Bunse-Gerstner and C. He, *On a Sturm sequence of polynomials for unitary Hessenberg matrices*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1043–1055.

[11] P. D. Dragnev and E. B. Saff, *Constrained energy problems with applications to orthogonal polynomials of a discrete variable*, J. Anal. Math., 72 (1997), pp. 223–259.

[12] T. A. Driscoll, K.-C. Toh, and L. N. Trefethen, *From potential theory to matrix iterations in six steps*, SIAM Rev., 40 (1998), pp. 547–578.

[13] H. Fassbender, *Inverse unitary eigenproblems and related orthogonal functions*, Numer. Math., 77 (1997), pp. 323–345.

[14] W. B. Gragg, *The QR algorithm for unitary Hessenberg matrices*, J. Comput. Appl. Math., 16 (1986), pp. 1–8.

[15] 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.

[16] C. Jagels and L. Reichel, *The isometric Arnoldi process and an application to iterative solution of large linear systems*, in Iterative Methods in Linear Algebra, R. Beauwens and P. de Groen, eds., North-Holland, Amsterdam, 1992, pp. 361–369.

[17] W. B. Jones, O. Njåstad, and W. J. Thron, *Moment theory, orthogonal polynomials, quadrature, and continued fractions associated with the unit circle*, Bull. London Math. Soc., 21

(1989), pp. 113–152.
[18] A. B. J. Kuijlaars, *Which eigenvalues are found by the Lanczos method?*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 306–321.
[19] A. B. J. Kuijlaars and E. A. Rakhmanov, *Zero distributions for discrete orthogonal polynomials*, J. Comput. Appl. Math., 99 (1998), pp. 255–274.
[20] K. V. Mardia and P. E. Jupp, *Directional Statistics*, John Wiley and Sons, Chichester, UK, 2000.
[21] E. A. Rakhmanov, *Equilibrium measure and the distribution of zeros of the extremal polynomials of a discrete variable*, Sb. Math., 187 (1996), pp. 1213–1228.
[22] T. Ransford, *Potential Theory in the Complex Plane*, Cambridge University Press, Cambridge, UK, 1995.
[23] E. B. Saff and V. Totik, *Logarithmic Potentials with External Fields*, Springer-Verlag, Berlin, 1997.
[24] J. Shen, G. Strang, and A. J. Wathen, *The potential theory of several intervals and its applications*, Appl. Math. Optim., 44 (2001) 67–85.
[25] M. Stewart, *Stability properties of several variants of the unitary Hessenberg QR algorithm*, in Structured Matrices in Mathematics, Computer Science and Engineering, II, V. Olshevsky, ed., Contemp. Math. 281, AMS, Providence, RI, 2001, pp. 57–72.
[26] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 3, pp. 810–829

ORTHOGONAL RATIONAL FUNCTIONS AND STRUCTURED MATRICES*

MARC VAN BAREL†, DARIO FASINO‡, LUCA GEMIGNANI§, AND NICOLA MASTRONARDI¶

Abstract. The linear space of all proper rational functions with prescribed poles is considered. Given a set of points z_i in the complex plane and the weights w_i, we define the discrete inner product

$$\langle \phi, \psi \rangle := \sum_{i=0}^{n} |w_i|^2 \overline{\phi(z_i)} \psi(z_i).$$

In this paper we derive a method to compute the coefficients of a recurrence relation generating a set of orthonormal rational basis functions with respect to the discrete inner product. We will show that these coefficients can be computed by solving an inverse eigenvalue problem for a matrix having a specific structure. In the case where all the points z_i lie on the real line or on the unit circle, the computational complexity is reduced by an order of magnitude.

Key words. orthogonal rational functions, structured matrices, diagonal-plus-semiseparable matrices, inverse eigenvalue problems, recurrence relation

AMS subject classifications. 42C05, 65F18, 65D15

DOI. 10.1137/S0895479803444454

1. Introduction and motivation.

Proper rational functions are an essential tool in many areas of engineering, such as system theory and digital filtering, where polynomial models are inappropriate due to their unboundedness at infinity. In fact, for physical reasons the transfer functions describing linear time-invariant systems often have to be bounded on the real line. Furthermore, approximation problems with rational functions are in the core of, e.g., the partial realization problem [23], model reduction problems [6, 7, 13], and robust system identification [7, 27].

Recently a strong interest has been brought to a variety of rational interpolation problems where a given function is to be approximated by means of a rational function with prescribed poles (see [8, 9, 36] and the references given therein). By linearization, such problems naturally lead to linear algebra computations involving

*Received by the editors June 1, 2003; accepted for publication (in revised form) by G. H. Golub May 30, 2004; published electronically April 8, 2005.

http://www.siam.org/journals/simax/26-3/44445.html

†Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium (Marc.VanBarel@cs.kuleuven.ac.be). The research of this author was partially supported by the Fund for Scientific Research (FWO), "SMA: Structured Matrices and their Applications," grant G.0078.01, "ANCILA: Asymptotic Analysis of the Convergence Behavior of Iterative methods in numerical linear algebra," grant G.0176.02, the K. U. Leuven research project "SLAP: Structured Linear Algebra Package," grant OT-00-16, and the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology, and Culture, grant IPA V/22. The scientific responsibility rests with the author.

‡Dipartimento di Matematica e Informatica, Università degli Studi di Udine, Viale delle Scienze 208, 33100 Udine, Italy (fasino@dimi.uniud.it). The research of this author was partially supported by G.N.C.S. and MIUR, grant 2004015437.

§Dipartimento di Matematica, Università di Pisa, Via F. Buonarroti 2, 56127 Pisa, Italy (gemignan@dm.unipi.it). The research of this author was partially supported by G.N.C.S. and MIUR, grant 2004015437.

¶Istituto per le Applicazioni del Calcolo "M. Picone," sez. Bari, Consiglio Nazionale delle Ricerche, Via G. Amendola, 122/I, I-70126 Bari, Italy (N.Mastronardi@area.ba.cnr.it). The research of this author was partially supported by G.N.C.S. and MIUR, grant 2004015437.

structured matrices. Exploiting the close connections between the functional problem and its matrix counterparts generally allows us to take advantage of the special structure of these matrices to speed up the approximation scheme. For example, in [28] efficient algorithms are designed for rational function evaluation and interpolation from their connection with displacement structured matrices.

The purpose of this paper is to devise a procedure to construct a set of proper rational functions with prescribed poles that are orthogonal with respect to a discrete inner product. Orthogonal rational functions are useful in solving multipoint generalizations of classical moment problems and associated interpolation problems; see [9] for further references on this topic. We also mention the recent appearance in the numerical analysis literature of quadrature formulas that are exact for sets of rational functions having prescribed poles; see, e.g., [10, 21]. Such formulas provide a greater accuracy than standard quadrature formulas when the poles are chosen in such a way to mimic the poles present in the integrand. The construction of Gauss-type quadrature formulas is known to be a task closely related to that of orthogonalizing a set of prescribed basis functions. In the polynomial case this fact was explored by Reichel [29, 30] and Gragg and Harrod [24]. Indeed, in these papers the construction of polynomial sequences that are orthogonal with respect to a discrete inner product by means of their three-term recurrence relation is tied with the solution of an inverse eigenvalue problem for symmetric tridiagonal matrices that is equivalent to the construction of Gauss quadrature formulas [22].

In this paper we adapt the technique laid down in [30] for polynomial sequences to a specific set of proper rational functions. The goal is the computation of an orthonormal basis of the linear space $\mathcal{R}_n$ of proper rational functions $\phi(z) = n(z)/d(z)$ w.r.t. a discrete inner product $\langle \cdot, \cdot \rangle$. Here $\deg(n(z)) \leq \deg(d(z)) \leq n$ and $d(z)$ has a prescribed set $\{y_1, \ldots, y_n\}$, $y_i \in \mathbb{C}$, of possible zeros; moreover, we set $\langle \phi, \psi \rangle := \sum_{i=0}^{n} |w_i|^2 \overline{\phi(z_i)} \psi(z_i)$ for $\phi(z), \psi(z) \in \mathcal{R}_n$. Such computation arises in the solution of least squares approximation problems with rational functions with prescribed poles. Moreover, it is also closely related with the computation of an orthogonal factorization of Cauchy-like matrices whose nodes are the points z_i and y_i [16, 17].

We prove that an orthonormal basis of $(\mathcal{R}_n, \langle \cdot, \cdot \rangle)$ can be generated by means of a suitable recurrence relation. Fast $O(n^2)$ Stieltjes-like procedures for computing the coefficients of such a recurrence relation when the points z_i as well as the points y_i are all real were first devised in [16, 17]. However, like the polynomial (Vandermonde) case [29], these fast algorithms turn out to be quite sensitive to roundoff errors so that the computed functions are far from orthogonal. Therefore, in this paper we propose a different approach based on the reduction of the considered problem to the following inverse eigenvalue problem (DS-IEP): Find a matrix S of order $n+1$ whose lower triangular part is the lower triangular part of a rank 1 matrix, and a unitary matrix Q of order $n+1$ such that $Q^H \vec{w} = \|\vec{w}\| \vec{e}_1$ and $Q^H D_z Q = S + D_y$. Here and below $\vec{w} = [w_0, \ldots, w_n]^T$, $D_z = \mathrm{diag}[z_0, \ldots, z_n]$, and $D_y = \mathrm{diag}[y_0, \ldots, y_n]$, where y_0 can be chosen arbitrarily. Moreover, we denote by $\mathcal{S}_k$ the class of $k \times k$ matrices S whose lower triangular part is the lower triangular part of a rank 1 matrix. If both S and S^H belong to $\mathcal{S}_k$, then S is called a *semiseparable* matrix [32]. Symmetric semiseparable matrices have also been called matrices *à la couple* by Gantmacher and Krein [20] and are special instances of *Green* matrices, which were defined by Asplund [3] with the aim of describing the inverses of band matrices.

A quite similar reduction to an inverse eigenvalue problem for a tridiagonal symmetric matrix (T-IEP) or for a unitary Hessenberg matrix (H-IEP) was also exploited in the theory on the construction of orthonormal polynomials w.r.t. a discrete inner

product (see [2, 5, 15, 19, 24, 29, 31, 33] for a survey of the theory and applications on T-IEP and H-IEP). This theory can be generalized to orthonormal *vector* polynomials; we refer the interested reader to [1, 11, 34, 35, 36, 37]. Since invertible semiseparable matrices are the inverses of tridiagonal ones [3, 20], we find that DS-IEP gives a generalization of T-IEP and, in particular, it reduces to T-IEP in the case where $y_i, z_i \in \mathbb{R}$ and all prescribed poles y_i are equal.

We devise a method for solving DS-IEP which fully exploits its recursive properties. This method proceeds by applying a sequence of carefully chosen Givens rotations to update the solution at the kth step by adding new data $(w_{k+1}, z_{k+1}, y_{k+1})$. The unitary matrix Q can thus be determined in its factored form as a product of $O(n^2)$ Givens rotations at the cost of $O(n^2)$ arithmetic operations (ops). The complexity of forming the matrix S depends on the structural properties of its upper triangular part and, in general, it requires $O(n^3)$ ops. In the case where all the points z_i lie on the real axis, we show that S is a semiseparable matrix so that the computation of S can be carried out using $O(n^2)$ ops only. In addition to that, the class $\mathcal{S}_{n+1}$ results to be close under linear fractional (Moebius) transformations of the form $z \to (\alpha z + \beta)/(\gamma z + \delta)$. Hence, by combining these two facts together, we are also able to prove that the process of forming S can be performed at the cost of $O(n^2)$ ops whenever all points z_i belong to a generalized circle (ordinary circles and straight lines) in the complex plane.

This paper is organized in the following way. In section 2 we reduce the computation of a sequence of orthonormal rational basis functions to the solution of an inverse eigenvalue problem for matrices of the form $\mathrm{diag}[y_0, \ldots, y_n] + S$, with $S \in \mathcal{S}_{n+1}$. By exploiting this reduction, we also determine relations for the recursive construction of such functions. Section 3 provides our method for solving DS-IEP in the general case whereas the more specific situations corresponding to points lying on the real axis on the unit circle or on a generic circle in the complex plane are considered in section 4. Finally, in section 5 we discuss numerical and computational issues related to the practical application of our method for solving DS-IEP by computations in fixed precision floating-point arithmetic.

2. The computation of orthonormal rational functions and its matrix framework. In this section we will study the properties of a sequence of proper rational functions with prescribed poles that are orthonormal with respect to a certain discrete inner product. We will also design an algorithm to compute such a sequence via a suitable recurrence relation. The derivation of this algorithm follows from reducing the functional problem into a matrix setting to the solution of an inverse eigenvalue problem involving structured matrices.

2.1. The functional problem. Given the complex numbers $y_1, y_2, \ldots, y_n$ all different from each other, let us consider the vector space $\mathcal{R}_n$ of all proper rational functions having possible poles in $y_1, y_2, \ldots, y_n$:

$$\mathcal{R}_n := \mathrm{span}\left\{1, \frac{1}{z-y_1}, \frac{1}{z-y_2}, \ldots, \frac{1}{z-y_n}\right\}.$$

The vector space $\mathcal{R}_n$ can be equipped with the inner product $\langle \cdot, \cdot \rangle$ defined below.

DEFINITION 2.1 (bilinear form). *Given the complex numbers $z_0, z_1, \ldots, z_n$, which together with the numbers y_i are all different from each other, and the "weights"*

$0 \neq w_i$, $i = 0, 1, \ldots, n$, *we define a bilinear form* $\langle \cdot, \cdot \rangle : \mathcal{R}_n \times \mathcal{R}_n \to \mathbb{C}$ *by*

$$\langle \phi, \psi \rangle := \sum_{i=0}^{n} |w_i|^2 \overline{\phi(z_i)} \psi(z_i).$$

Since there is no proper rational function $\phi(z) = n(z)/d(z)$ with $\deg(n(z)) \leq \deg(d(z)) \leq n$ different from the zero function such that $\phi(z_i) = 0$ for $i = 0, \ldots, n$, this bilinear form defines a positive definite inner product in the space $\mathcal{R}_n$.

The aim of this paper is to develop an efficient algorithm for the solution of the following functional problem.

PROBLEM 1 (computing a sequence of orthonormal rational basis functions). *Construct an orthonormal basis*

$$\vec{\alpha}_n(z) := [\alpha_0(z), \alpha_1(z), \ldots, \alpha_n(z)]$$

of $(\mathcal{R}_n, \langle \cdot, \cdot \rangle)$ *satisfying the properties*

$$\begin{aligned} \alpha_j(z) &\in \mathcal{R}_j \setminus \mathcal{R}_{j-1} \qquad (\mathcal{R}_{-1} := \emptyset), \\ \langle \alpha_i, \alpha_j \rangle &= \delta_{i,j} \qquad (\textit{Kronecker delta}) \end{aligned}$$

for $i, j = 0, 1, 2, \ldots, n$.

Later, we will show that the computation of such an orthonormal basis $\vec{\alpha}_n(z)$ is equivalent to the solution of an inverse eigenvalue problem for matrices of the form $\mathrm{diag}[y_0, \ldots, y_n] + S$, where $S \in \mathcal{S}_{n+1}$.

2.2. The inverse eigenvalue problem. Let $D_y = \mathrm{diag}[y_0, \ldots, y_n]$ be the diagonal matrix whose diagonal elements are $y_0, y_1, \ldots, y_n$, where y_0 can be chosen arbitrarily; analogously, set $D_z = \mathrm{diag}[z_0, \ldots, z_n]$. Recall that $\mathcal{S}_k$ is the class of $k \times k$ matrices S whose lower triangular part is the lower triangular part of a rank 1 matrix. Furthermore, denote by $\|\vec{w}\|$ the Euclidean norm of the vector $\vec{w} = [w_0, w_1, \ldots, w_n]^T$.

Our approach to solving Problem 1 mainly relies upon the equivalence between that problem and the following inverse eigenvalue problem for diagonal-plus-semi separable matrices (DS-IEP).

PROBLEM 2 (solving an inverse eigenvalue problem). *Given the numbers* w_i, z_i, y_i, *find a matrix* $S \in \mathcal{S}_{n+1}$ *and a unitary matrix* Q *such that*

$$Q^H \vec{w} = \|\vec{w}\| \vec{e}_1, \tag{2.1}$$

$$Q^H D_z Q = S + D_y. \tag{2.2}$$

Remark 1. Observe that if (Q, S) is a solution of Problem 2, then S cannot have zero rows and columns. By contradiction, if we suppose that $S\vec{e}_j = \vec{0}$, where $\vec{e}_j$ is the jth column of the identity matrix I_{n+1} of order $n+1$, then $D_z Q \vec{e}_j = Q D_y \vec{e}_j = y_{j-1} Q \vec{e}_j$, from which it would follow $y_{j-1} = z_i$ for a certain i.

Results concerning the existence and uniqueness of the solution of Problem 2 were first proven in the papers [16, 17, 18] for the specific case where $y_i, z_i \in \mathbb{R}$ and S is a semiseparable matrix. In particular, under such auxiliary assumptions, it was shown that the matrix Q is simply the orthogonal factor of a QR decomposition of a Cauchy-like matrix built from the nodes y_i and z_i, i.e., a matrix whose (i, j)th element has the form $u_{i-1} v_{j-1} / (z_{i-1} - y_{j-1})$ where u_{i-1} and v_{j-1} are components of two vectors $\vec{u} = [u_0, \ldots, u_n]^T$ and $\vec{v} = [v_0, \ldots, v_n]^T$, respectively. Next, we give a generalization of the results of [16, 17, 18] to deal with the more general situation considered here.

THEOREM 2.2. *Problem* 2 *has at least one solution. If* (Q_1, S_1) *and* (Q_2, S_2) *are two solutions of Problem* 2*, then there exists a unitary diagonal matrix* $F = \mathrm{diag}[1, \mathrm{e}^{\mathrm{i}\theta_1}, \ldots, \mathrm{e}^{\mathrm{i}\theta_n}]$ *such that*

$$Q_2 = Q_1 F, \quad S_2 = F^H S_1 F.$$

Proof. It is certainly possible to find two vectors $\vec{u} = [u_0, \ldots, u_n]^T$ and $\vec{v} = [v_0, \ldots, v_n]^T$ with $v_i, u_i \neq 0$ and $u_i v_0/(z_i - y_0) = w_i$, for $0 \leq i \leq n$. Indeed, it is sufficient to set, for example, $v_i = 1$ and $u_i = w_i(z_i - y_0)$. Hence, let us consider the nonsingular Cauchy-like matrix $C \equiv (u_{i-1}v_{j-1}/(z_{i-1} - y_{j-1}))$ and let $C = QR$ be a QR factorization of C. From $D_z C - C D_y = \vec{u}\vec{v}^T$ one easily finds that

$$Q^H D_z Q = R D_y R^{-1} + Q\vec{u}\vec{v}^T R^{-1} = D_y + S,$$

where

$$S = R D_y R^{-1} - D_y + Q\vec{u}\vec{v}^T R^{-1} \in \mathcal{S}_{n+1}.$$

Moreover, $Q\vec{e}_1 = CR^{-1}\vec{e}_1 = \vec{w}/\|\vec{w}\|$ by construction. Hence, the matrices Q and $S = Q^H D_z Q - D_y$ solve Problem 2.

Concerning uniqueness, assume that (Q, S) is a solution of Problem 2 with $S \equiv (s_{i,j})$ and $s_{i,j} = \widetilde{u}_{i-1}\widetilde{v}_{j-1}$ for $1 \leq j \leq i \leq n+1$. As $S\vec{e}_1 \neq \vec{0}$, it follows that $\widetilde{v}_0 \neq 0$ and, therefore, we may assume $\widetilde{v}_0 = 1$. Moreover, from (2.2) it is easily found that

$$D_z Q\vec{e}_1 = Q\vec{\widetilde{u}} + y_0 Q\vec{e}_1,$$

where $\vec{\widetilde{u}} = [\widetilde{u}_0, \ldots, \widetilde{u}_n]^T$. From (2.1) we have

$$\vec{\widetilde{u}} = Q^H (D_z - y_0 I_{n+1}) \frac{\vec{w}}{\|\vec{w}\|}. \tag{2.3}$$

Relation (2.2) can be rewritten as

$$Q^H D_z Q = \vec{\widetilde{u}}\vec{\widetilde{v}}^T + U = \vec{\widetilde{u}}\vec{\widetilde{v}}^T + R D_y R^{-1},$$

where U is an upper triangular matrix with diagonal entries y_i and $U = R D_y R^{-1}$ gives its Jordan decomposition, defined up to a suitable scaling of the columns of the upper triangular eigenvector matrix R. Hence, we find that

$$D_z QR - QRD_y = Q\vec{\widetilde{u}}\vec{\widetilde{v}}^T R = \vec{u}\vec{v}^T$$

and, therefore, $QR = C \equiv (u_{i-1}v_{j-1}/(z_{i-1} - y_{j-1}))$ is a Cauchy-like matrix with $\vec{u} = Q\vec{\widetilde{u}}$ uniquely determined by (2.3). This means that all the eligible Cauchy-like matrices C are obtained one from the other by a post-multiplication side by a suitable diagonal matrix. In this way, from the essential uniqueness of the orthogonal factorization of a given matrix, we may conclude that Q is uniquely determined up to post-multiplication side by a unitary diagonal matrix F having fixed its first diagonal entry equal to 1. Finally, the result for S immediately follows from using again relation (2.2). ∎

The above theorem says that the solution of Problem 2 is essentially unique up to a diagonal scaling. Furthermore, once the weight vector $\vec{w}$ and the points z_i are fixed, the determinant of S is a rational function in the variables $y_0, \ldots, y_n$ whose

numerator is not identically zero. Hence, we can show that for almost any choice of $y_0, \ldots, y_n$, the resulting matrix S is nonsingular. The paper [16] dealt with this *regular* case in the framework of the orthogonal factorization of real Cauchy matrices. In particular, it is shown there that the matrix S is nonsingular when all the nodes y_i, z_i are real and there exists an interval, either finite or infinite, containing all nodes y_i and none of the nodes z_i.

In what follows we assume that $S^{-1} = H$ exists. It is known that the inverse of a matrix whose lower triangular part is the lower triangular part of a rank 1 matrix is an irreducible Hessenberg matrix [20]. Hence, we will use the following notation: The matrix $H = S^{-1}$ is upper Hessenberg with subdiagonal elements $b_0, b_1, \ldots, b_{n-1}$; for $j = 0, \ldots, n-1$, the jth column H_j of H has the form

$$H_j^T =: [\vec{h}_j^T, b_j, \vec{0}], \quad b_j \neq 0.$$

The outline of the remainder of this section is as follows. First we assume that we know a unitary matrix Q and the corresponding matrix S solving Problem 2. Then we provide a recurrence relation between the columns Q_j of Q and, in addition to that, we give a connection between the columns Q_j and the values at the points z_i attained by certain rational functions satisfying a similar recurrence relation. Finally, we show that these rational functions form a basis we are looking for.

2.3. Recurrence relation for the columns of Q. Let the columns of Q denote as follows:

$$Q =: [Q_0, Q_1, \ldots, Q_n].$$

THEOREM 2.3 (recurrence relation). *For $j = 0, 1, \ldots, n$, the columns Q_j satisfy the recurrence relation*

$$b_j(D_z - y_{j+1} I_{n+1}) Q_{j+1} = Q_j + ([Q_0, Q_1, \ldots, Q_j]\, D_{y,j} - D_z\, [Q_0, Q_1, \ldots, Q_j])\, \vec{h}_j,$$

with $Q_0 = \vec{w}/\|\vec{w}\|$, $Q_{n+1} = 0$, and $D_{y,j} = \operatorname{diag}[y_0, \ldots, y_j]$.

Proof. Since $Q^H \vec{w} = \vec{e}_1 \|\vec{w}\|$, it follows that $Q_0 = \vec{w}/\|\vec{w}\|$. Multiplying relation (2.2) to the left by Q, we have

$$D_z Q = Q(S + D_y).$$

Post-multiplying this by $H = S^{-1}$ gives us

$$D_z Q H = Q(I_{n+1} + D_y H). \tag{2.4}$$

Considering the jth column of the left- and right-hand sides of the equation above we have the claim. ∎

2.4. Recurrence relation for the orthonormal rational functions. In this section we define an orthonormal basis $\vec{\alpha}_n(z) = [\alpha_0(z), \alpha_1(z), \ldots, \alpha_n(z)]$ for $\mathcal{R}_n$ using a recurrence relation built by means of the information contained in the matrix H.

DEFINITION 2.4 (recurrence for the orthonormal rational functions). *Let us define $\alpha_0(z) = 1/\|\vec{w}\|$ and*

$$\alpha_{j+1}(z) = \frac{\alpha_j(z) + ([\alpha_0(z), \ldots, \alpha_j(z)]\, D_{y,j} - z\, [\alpha_0(z), \ldots, \alpha_j(z)])\, \vec{h}_j}{b_j(z - y_{j+1})}$$

for $0 \leq j \leq n-1$.

In the next theorem, we prove that the rational functions $\alpha_j(z)$ evaluated at the points z_i are connected to the elements of the unitary matrix Q. This will allow us to prove in Theorem 2.6 that the rational functions $\alpha_j(z)$ are indeed the orthonormal rational functions we are looking for. In what follows, we use the notation $D_w = \mathrm{diag}[w_0, \ldots, w_n]$.

THEOREM 2.5 (connection between $\alpha_j(z_i)$ and the elements of Q). *Let*

$$\vec{\alpha}_j = [\alpha_j(z_0), \ldots, \alpha_j(z_n)]^T \in \mathbb{C}^{n+1}, \quad 0 \le j \le n.$$

For $j = 0, 1, \ldots, n$, we have $Q_j = D_w \vec{\alpha}_j$.

Proof. Replacing z by z_i in the recurrence relation for $\alpha_{j+1}(z)$, we get

$$b_j(D_z - y_{j+1} I_{n+1})\vec{\alpha}_{j+1} = \vec{\alpha}_j + ([\vec{\alpha}_0, \ldots, \vec{\alpha}_j]\, D_{y,j} - D_z\, [\vec{\alpha}_0, \ldots, \vec{\alpha}_j])\, \vec{h}_j.$$

Since $Q_0 = \vec{w}/\|\vec{w}\| = D_w \vec{\alpha}_0$, the theorem is proved by finite induction on j, comparing the preceding recurrence with the one in Theorem 2.3. ∎

Now it is easy to prove the orthonormality of the rational functions $\alpha_j(z)$.

THEOREM 2.6 (orthonormality of $\vec{\alpha}_n(z)$). *The functions $\alpha_0(z), \ldots, \alpha_n(z)$ form an orthonormal basis for $\mathcal{R}_n$ with respect to the inner product $\langle \cdot, \cdot \rangle$. Moreover, we have $\alpha_j(z) \in \mathcal{R}_j \setminus \mathcal{R}_{j-1}$.*

Proof. First we prove that $\langle \alpha_i, \alpha_j \rangle = \delta_{i,j}$. This follows immediately from the fact that $Q = D_w[\vec{\alpha}_0, \ldots, \vec{\alpha}_n]$ and Q is unitary. Now we have to prove that $\alpha_j(z) \in \mathcal{R}_j \setminus \mathcal{R}_{j-1}$. This is clearly true for $j = 0$ (recall that $\mathcal{R}_{-1} = \emptyset$). Suppose it is true for $j = 0, 1, 2, \ldots, k < n$. From the recurrence relation, we derive that $\alpha_{k+1}(z)$ has the form

$$\alpha_{k+1}(z) = \frac{\text{rational function with possible poles in } y_0, y_1, \ldots, y_k}{(z - y_{j+1})}.$$

Also $\lim_{z \to \infty} \alpha_{k+1}(z) \in \mathbb{C}$ and, therefore, $\alpha_{k+1}(z) \in \mathcal{R}_{k+1}$. Note that simplification by $(z - y_{k+1})$ does not occur in the previous formula for $\alpha_{k+1}(z)$ because $Q_{k+1} = D_w \vec{\alpha}_{k+1}$ is linearly independent of the previous columns of Q. Hence, $\alpha_{k+1}(z) \in \mathcal{R}_{k+1} \setminus \mathcal{R}_k$. ∎

In the next theorem, we give an alternative relation among the rational functions $\alpha_j(z)$.

THEOREM 2.7 (alternative relation). *We have*

$$z\vec{\alpha}_n(z) = \vec{\alpha}_n(z)(S + D_y) + \alpha_{n+1}(z)\vec{s}_n, \tag{2.5}$$

where $\vec{s}_n$ is the last row of the matrix S and the function $\alpha_{n+1}(z)$ is given by

$$\alpha_{n+1}(z) = c \prod_{j=0}^{n} (z - z_j) / \prod_{j=1}^{n} (z - y_j)$$

for some constant c.

Proof. Let H_n be the last column of $H = S^{-1}$ and define

$$\alpha_{n+1}(z) = \vec{\alpha}_n(z)(zI_{n+1} - D_y)H_n - \alpha_n(z). \tag{2.6}$$

Thus, the recurrence relation given in Definition 2.4 can also be written as

$$\vec{\alpha}_n(z)(zI_{n+1} - D_y)H = \vec{\alpha}_n(z) + \alpha_{n+1}(z)\vec{e}_{n+1}^T.$$

Post-multiplying by $S = H^{-1}$, we obtain the formula (2.5). To determine the form of $\alpha_{n+1}(z)$ we look at the definition (2.6). It follows that $\alpha_{n+1}(z)$ is a rational function having degree of numerator at most one more than the degree of the denominator and having possible poles in $y_1, y_2, \ldots, y_n$. Recalling from Theorem 2.5 the notation $\vec{\alpha}_j = [\alpha_j(z_0), \ldots, \alpha_j(z_n)]^T$ and the equation $Q = D_w[\vec{\alpha}_0, \ldots, \vec{\alpha}_n]$, we can evaluate the previous equation at the points z_i and obtain

$$D_z\,[\vec{\alpha}_0, \ldots, \vec{\alpha}_n]\,H - [\vec{\alpha}_0, \ldots, \vec{\alpha}_n]\,D_y H = [\vec{\alpha}_0, \ldots, \vec{\alpha}_n] + \vec{\alpha}_{n+1}\vec{e}_n^T.$$

Since $D_w D_z = D_z D_w$, pre-multiplying by D_w we obtain

$$D_z Q H - Q D_y H = Q + D_w \vec{\alpha}_{n+1}\vec{e}_{n+1}^T.$$

From (2.4) we obtain that $D_w\vec{\alpha}_{n+1}\vec{e}_{n+1}^T$ is a zero matrix; hence, it follows that $\alpha_{n+1}(z_i) = 0$, for $i = 0, 1, \ldots, n$, and this proves the theorem. ∎

Note that $\alpha_{n+1}(z)$ is orthogonal to all $\alpha_i(z)$, $i = 0, 1, \ldots, n$, since $\alpha_{n+1}(z) \notin \mathcal{R}_n$ and its norm is

$$\|\alpha_{n+1}\|^2 = \sum_{i=0}^{n} |w_i \alpha_{n+1}(z_i)|^2 = 0.$$

3. Solving the inverse eigenvalue problem. In this section we devise an efficient recursive procedure for the construction of the matrices Q and S solving Problem 2 (DS-IEP). The case $n = 0$ is trivial; it is sufficient to set $Q = w_0/|w_0|$ and $S = z_0 - y_0$. Let us assume we have already constructed a unitary matrix Q_k and a matrix S_k for the first $k+1$ points $z_0, z_1, \ldots, z_k$ with the corresponding weights $w_0, w_1, \ldots, w_k$. That is, (Q_k, S_k) satisfies

$$\begin{aligned} Q_k^H \vec{w}_k &= \|\vec{w}_k\|\vec{e}_1, \\ Q_k^H D_{z,k} Q_k &= S_k + D_{y,k}, \end{aligned}$$

where $\vec{w}_k = [w_0, \ldots, w_k]^T$, $S_k \in \mathcal{S}_{k+1}$, $D_{z,k} = \operatorname{diag}[z_0, \ldots, z_k]$, and, similarly, $D_{y,k} = \operatorname{diag}[y_0, \ldots, y_k]$. The idea is now to add a new point z_{k+1} with corresponding weight w_{k+1} and construct the corresponding matrices Q_{k+1} and S_{k+1}.

Hence, we start with the following relations:

$$\begin{bmatrix} 1 & 0 \\ 0 & Q_k^H \end{bmatrix} \begin{bmatrix} w_{k+1} \\ \vec{w}_k \end{bmatrix} = \begin{bmatrix} w_{k+1} \\ \|\vec{w}_k\|\vec{e}_1 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 0 \\ 0 & Q_k^H \end{bmatrix} \begin{bmatrix} z_{k+1} & 0 \\ 0 & D_{z,k} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & Q_k \end{bmatrix} = \begin{bmatrix} z_{k+1} & 0 \\ 0 & S_k + D_{y,k} \end{bmatrix}.$$

Then, we find complex Givens rotations $G_i = I_{i-1} \oplus G_{i,i+1} \oplus I_{k-i+1}$,

$$G_{i,i+1} =: \begin{bmatrix} c & s \\ -\overline{s} & c \end{bmatrix}, \quad G_{i,i+1}^H G_{i,i+1} = I_2, \tag{3.1}$$

such that

$$G_k^H, \ldots, G_1^H \begin{bmatrix} 1 & 0 \\ 0 & Q_k^H \end{bmatrix} \begin{bmatrix} w_{k+1} \\ \vec{w}_k \end{bmatrix} = \begin{bmatrix} \|\vec{w}_{k+1}\| \\ 0 \end{bmatrix}$$

and, moreover,

$$G_k^H, \ldots, G_1^H \begin{bmatrix} 1 & 0 \\ 0 & Q_k^H \end{bmatrix} \begin{bmatrix} z_{k+1} & 0 \\ 0 & D_{z,k} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & Q_k \end{bmatrix} G_1, \ldots, G_k \in \mathcal{S}_{k+2}.$$

Finally, we set

$$Q_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & Q_k \end{bmatrix} G_1, \ldots, G_k$$

and

$$S_{k+1} = G_k^H, \ldots, G_1^H \begin{bmatrix} z_{k+1} & 0 \\ 0 & S_k + D_{y,k} \end{bmatrix} G_1, \ldots, G_k. \tag{3.2}$$

With the notation

$$SS\left(\begin{matrix} [u_0, u_1, \ldots, u_k] \\ [v_0, v_1, \ldots, v_k] \end{matrix} \right)$$

we denote the lower triangular matrix whose nonzero part equals the lower triangular part of the rank 1 matrix $[u_{i-1}v_{j-1}]_{i=0,\ldots,k}^{j=0,\ldots,k}$. Moreover, with the notation

$$RR\left(\begin{matrix} [\eta_0, \eta_1, \ldots, \eta_{k-1}] \\ [\vec{r}_0, \vec{r}_1, \ldots, \vec{r}_{k-2}] \end{matrix} \right)$$

we denote the strictly upper triangular matrix whose $(i+1)$st row, $0 \le i \le k-2$, is equal to $[\vec{0}, \eta_i, \vec{r}_i^T]$.

Let us now describe in what way Givens rotations are selected in order to perform the updating of Q_k and S_k. In the first step we construct a Givens rotation working on the new weight. Let $G_{1,2}$ be a Givens rotation as in (3.1), such that

$$G_{1,2}^H \begin{bmatrix} w_{k+1} \\ \|\vec{w}_k\| \end{bmatrix} = \begin{bmatrix} \|\vec{w}_{k+1}\| \\ 0 \end{bmatrix}. \tag{3.3}$$

The matrix S_k is updated as follows. We know that

$$S_k = SS\left(\begin{matrix} [u_0, u_1, \ldots, u_k] \\ [v_0, v_1, \ldots, v_k] \end{matrix} \right) + RR\left(\begin{matrix} [\eta_0, \eta_1, \ldots, \eta_{k-1}] \\ [\vec{r}_0, \vec{r}_1, \ldots, \vec{r}_{k-2}] \end{matrix} \right).$$

Let

$$S_{k+1,1} + D_{y,k+1,1} := \begin{bmatrix} G_{1,2}^H & 0 \\ 0 & I_k \end{bmatrix} \begin{bmatrix} z_{k+1} & 0 \\ 0 & S_k + D_{y,k} \end{bmatrix} \begin{bmatrix} G_{1,2} & 0 \\ 0 & I_k \end{bmatrix},$$

where $S_{k+1,1}$ and $D_{y,k+1,1}$ are defined as follows:

$$S_{k+1,1} = SS\left(\begin{matrix} [\hat{u}_0, \tilde{u}_1, u_1, u_2, \ldots, u_k] \\ [\hat{v}_0, \tilde{v}_1, v_1, v_2, \ldots, v_k] \end{matrix} \right) + RR\left(\begin{matrix} [\hat{\eta}_0, \tilde{\eta}_1, \eta_1, \ldots, \eta_{k-1}] \\ [\vec{\hat{r}}_0, \vec{\tilde{r}}_1, \vec{r}_1 \ldots, \vec{r}_{k-2}] \end{matrix} \right)$$

and

$$D_{y,k+1,1} = \operatorname{diag}(y_0, \tilde{y}_1, y_1, y_2, \ldots, y_k),$$

with

$$\begin{bmatrix} \alpha & \delta \\ \gamma & \beta \end{bmatrix} := G_{1,2}^H \begin{bmatrix} z_{k+1} & 0 \\ 0 & y_0 + u_0 v_0 \end{bmatrix} G_{1,2}$$

and

$$\begin{array}{lll} \hat{v}_0 = -\overline{s}v_0, & \hat{u}_0 = (\alpha - y_0)/\hat{v}_0, & \hat{\eta}_0 = \delta, \\ \tilde{v}_1 = cv_0, & \tilde{y}_1 = \beta - \tilde{u}_1\tilde{v}_1, & \tilde{u}_1 = \gamma/\hat{v}_0, \\ \tilde{\eta}_1 = c\eta_0, & \vec{\tilde{r}}_0 = \left[-s\eta_0, -s\vec{r}_0^{\,T}\right]^T, & \vec{\tilde{r}}_1 = c\vec{r}_0. \end{array}$$

Observe that $v_0 \neq 0$ from Remark 1 and, moreover, $s \neq 0$ since $\|\vec{w}_k\| \neq 0$ in (3.3), whence $\hat{v}_0 \neq 0$ and, therefore, all these quantities are well defined.

In the next steps, we are transforming $D_{y,k+1,1}$ into $D_{y,k+1}$. The first of these steps is as follows. If $v_1\tilde{u}_1 - \tilde{\eta}_1 \neq 0$, we choose t such that

$$\bar{t} = \frac{y_1 - \tilde{y}_1}{v_1\tilde{u}_1 - \tilde{\eta}_1}$$

and define the Givens rotation working on the 2nd and 3rd rows and columns as

$$G_{2,3} = \begin{bmatrix} 1 & t \\ -\bar{t} & 1 \end{bmatrix} / \sqrt{1 + |t|^2}.$$

Otherwise, if $v_1\tilde{u}_1 - \tilde{\eta}_1 = 0$, we set

$$G_{2,3} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

The matrices $S_{k+1,1}$ and $D_{y,k+1,1}$ undergo the similarity transformation associated to $G_{2,3}$. The transformed matrices $S_{k+1,2}$ and $D_{y,k+1,2}$ are given by

$$S_{k+1,2} = SS\begin{pmatrix} [\hat{u}_0, \hat{u}_1, \tilde{u}_2, u_2, \ldots, u_k] \\ [\hat{v}_0, \hat{v}_1, \tilde{v}_2, v_2, \ldots, v_k] \end{pmatrix} + RR\begin{pmatrix} [\hat{\eta}_0, \hat{\eta}_1, \tilde{\eta}_2, \eta_2, \ldots, \eta_{k-1}] \\ [\vec{\hat{r}}_0, \vec{\hat{r}}_1, \vec{\tilde{r}}_2, \vec{r}_2 \ldots, \vec{r}_{k-2}] \end{pmatrix},$$

$$D_{y,k+1,2} = \operatorname{diag}(y_0, y_1, \tilde{y}_2, y_2, y_3, \ldots, y_k),$$

with

$$G_{2,3}^H \begin{bmatrix} \tilde{u}_1 \\ u_1 \end{bmatrix} = \begin{bmatrix} \hat{u}_1 \\ \tilde{u}_2 \end{bmatrix}, \quad [\tilde{v}_1, v_1]\, G_{2,3} = [\hat{v}_1, \tilde{v}_2]\,.$$

Moreover, $\tilde{y}_2 = \tilde{y}_1$, $\hat{\eta}_1$ is the $(1,2)$-entry of

$$G_{2,3}^H \begin{bmatrix} \tilde{u}_1\tilde{v}_1 + \tilde{y}_1 & \tilde{\eta}_1 \\ u_1\tilde{v}_1 & u_1v_1 + y_1 \end{bmatrix} G_{2,3},$$

and

$$G_{2,3}^H \begin{bmatrix} \vec{\tilde{r}}_1 \\ [\eta_1, \vec{r}_1] \end{bmatrix} = \begin{bmatrix} \vec{\hat{r}}_1 \\ [\tilde{\eta}_2, \vec{\tilde{r}}_2] \end{bmatrix}. \tag{3.4}$$

At the very end, after k steps, we obtain

$$S_{k+1,k} = SS\begin{pmatrix} [\hat{u}_0, \hat{u}_1, \ldots, \hat{u}_k, \tilde{u}_{k+1}] \\ [\hat{v}_0, \hat{v}_1, \ldots, \hat{v}_k, \tilde{v}_{k+1}] \end{pmatrix} + RR\begin{pmatrix} [\hat{\eta}_0, \hat{\eta}_1, \hat{\eta}_2, \ldots, \hat{\eta}_k] \\ [\vec{\hat{r}}_0, \vec{\hat{r}}_1, \vec{\hat{r}}_2, \ldots, \vec{\hat{r}}_{k-1}] \end{pmatrix}$$

and

$$D_{y,k+1,k} = \operatorname{diag}(y_0, y_1, \ldots, y_k, \tilde{y}_{k+1}).$$

Since $S_{k+1,k} = S_{k+1} + \mathrm{diag}[0, \ldots, 0, \tilde{y}_{k+1} - y_{k+1}]$, then from (3.2) it follows that $\tilde{u}_{k+1} \neq 0$. Thus, the last step will transform $\tilde{y}_{k+1}$ into y_{k+1} by applying the transformation

$$\begin{aligned} \hat{u}_{k+1} &\leftarrow \tilde{u}_{k+1}, \\ \hat{v}_{k+1} &\leftarrow (\tilde{y}_{k+1} - y_{k+1} + \tilde{u}_{k+1}\tilde{v}_{k+1})/\tilde{u}_{k+1}. \end{aligned}$$

The computational complexity of the algorithm is dominated by the cost of performing the multiplications (3.4). In general, adding new data $(w_{k+1}, z_{k+1}, y_{k+1})$ requires $\mathcal{O}(k^2)$ ops and, hence, computing $S_n = S$ requires $\mathcal{O}(n^3)$ ops. In the next section we will show that these estimates reduce by an order of magnitude in the case where some special distributions of the points z_i are considered which lead to a matrix S with a structured upper triangular part. We stress the fact that in light of Theorem 2.2, the above procedure to solve DS-IEP can also be seen as a method to compute the orthogonal factor in a QR factorization of a suitable Cauchy-like matrix.

4. Special configurations of points z_i. In this section we specialize our algorithm for the solution of DS-IEP to cover with the important case where the points z_i are assumed to lie on the real axis or on the unit circle in the complex plane. Under this assumption on the distribution of the points z_i, it will be shown that the matrix S also possesses a semiseparable structure. The exploitation of this property allows us to overcome the multiplication (3.4) and to construct the matrix $S_n = S$ by means of a simpler parametrization, using $\mathcal{O}(n)$ ops per point, so that the overall cost of forming S reduces to $\mathcal{O}(n^2)$ ops.

4.1. Special case: All points z_i are real. When all the points z_i are real, we have

$$S + D_y = Q^H D_z Q = (Q^H D_z Q)^H = (S + D_y)^H.$$

Hence, the matrix $S + D_y$ can be written as

$$S + D_y = \mathrm{tril}(\vec{u}\vec{v}^T, 0) + D_y + \mathrm{triu}(\bar{\vec{v}}\vec{u}^H, 1), \tag{4.1}$$

with $\bar{\vec{v}}$ the complex conjugate of the vector $\vec{v}$. Here we adopt the Matlab notation $\mathrm{triu}(B, p)$ for the upper triangular portion of a square matrix B, where all entries below the pth diagonal are set to zero ($p = 0$ is the main diagonal, $p > 0$ is above the main diagonal, and $p < 0$ is below the main diagonal). Analogously, the matrix $\mathrm{tril}(B, p)$ is formed by the lower triangular portion of B by setting to zero all its entries above the pth diagonal. In particular, the matrix S is a Hermitian semiseparable matrix and its computation requires only $\mathcal{O}(n)$ ops per point, since its upper triangular part need not be computed via (3.4). Moreover, its inverse matrix H is tridiagonal; hence the vectors $\vec{h}_j$ occurring in Definition 2.4 have only one nonzero entry.

Also, when all the poles y_i (and the weights w_i) are real, all computations can be performed using real arithmetic instead of doing operations on complex numbers. When all the poles are real or come in complex conjugate pairs, all computations can also be done using only real arithmetic. However, the algorithm then works with a block diagonal D_y instead of a diagonal matrix. The details of this algorithm are rather elaborate so we will not go into the details here.

4.2. Special case: All points z_i lie on the unit circle. The case of points z_i located on the unit circle $\mathbb{T} = \{z \in \mathbb{C} : |z| = 1\}$ in the complex plane can be reduced to the real case treated in the preceding subsection by using the concept of linear

fractional (Moebius) transformation [25]. To be specific, a function $\mathcal{M} : \mathbb{C} \cup \{\infty\} \to \mathbb{C} \cup \{\infty\}$ is a Moebius transformation if

$$\mathcal{M}(z) = \frac{\alpha z + \beta}{\gamma z + \delta}, \quad \alpha\delta - \beta\gamma \neq 0, \qquad \alpha, \beta, \gamma, \delta \in \mathbb{C}.$$

Interesting properties concerning Moebius transformations are collected in [25]. In particular, a Moebius transformation defines a one-to-one mapping of the extended complex plane into itself and, moreover, the inverse of a Moebius transformation is still a Moebius transformation given by

$$\mathcal{M}^{-1}(z) = \frac{\delta z - \beta}{-\gamma z + \alpha}. \tag{4.2}$$

The Moebius transformation $\mathcal{M}(S)$ of a matrix S is defined as

$$\mathcal{M}(S) = (\alpha S + \beta I)(\gamma S + \delta I)^{-1}$$

if the matrix $\gamma S + \delta I$ is nonsingular. The basic fact relating semiseparable matrices with Moebius transformations is that in a certain sense the semiseparable structure is maintained under a Moebius transformation of the matrix. More precisely, we have the following theorem.

THEOREM 4.1. *Let $S \in \mathcal{S}_{n+1}$ with $S \equiv (s_{i,j})$, $s_{i,j} = u_{i-1}v_{j-1}$ for $1 \leq j \leq i \leq n+1$, and $v_0 \neq 0$. Moreover, let $D_y = \operatorname{diag}[y_0, \ldots, y_n]$ and assume that $\mathcal{M}$ maps the eigenvalues of both $S + D_y$ and D_y into points of the ordinary complex plane, i.e., $-\delta/\gamma$ is different from all the points y_i, z_i. Then, we find that*

$$\mathcal{M}(S + D_y) - \mathcal{M}(D_y) \in \mathcal{S}_{n+1}.$$

Proof. Observe that $S \in \mathcal{S}_{n+1}$ implies that $RSU \in \mathcal{S}_{n+1}$ for R and U upper triangular matrices. Hence, if we define $R = I - \vec{e}_1[0, v_1/v_0, \ldots, v_n/v_0]$, the theorem is proven by showing that

$$R^{-1}(\mathcal{M}(S + D_y) - \mathcal{M}(D_y))R \in \mathcal{S}_{n+1},$$

which is equivalent to

$$R^{-1}\mathcal{M}(S + D_y)R - \mathcal{M}(D_y) \in \mathcal{S}_{n+1}.$$

One immediately finds that

$$R^{-1}\mathcal{M}(S + D_y)R = ((\gamma(S + D_y) + \delta I)R)^{-1}(\alpha(S + D_y) + \beta I)R,$$

from which it follows that

$$R^{-1}\mathcal{M}(S + D_y)R = (\gamma v_0 \vec{u}\vec{e}_1^T + R_1)^{-1}(\alpha v_0 \vec{u}\vec{e}_1^T + R_2),$$

where R_1 and R_2 are upper triangular matrices with diagonal entries $\gamma y_i + \delta$ and $\alpha y_i + \beta$, respectively. In particular, R_1 is invertible and, by applying the Sherman–Morrison formula, we obtain

$$R^{-1}\mathcal{M}(S + D_y)R = (I - \sigma R_1^{-1}\vec{u}\vec{e}_1^T)(\alpha v_0 R_1^{-1}\vec{u}\vec{e}_1^T + R_1^{-1}R_2)$$

for a suitable σ. The thesis is now established by observing that the diagonal entries of $R_1^{-1}R_2$ coincide with the ones of $\mathcal{M}(D_y)$ and, moreover, from the previous relation one gets

$$R^{-1}\mathcal{M}(S+D_y)R - R_1^{-1}R_2 \in \mathcal{S}_{n+1}$$

and the proof is complete. ∎

This theorem has several interesting consequences since it is well known that we can determine Moebius transformations mapping the unit circle $\mathbb{T}$ except for one point onto the real axis in the complex plane. To see this, let us first consider Moebius transformations of the form

$$\mathcal{M}_1(z) = \frac{z+\bar{\alpha}}{z+\alpha}, \quad \alpha \in \mathbb{C} \setminus \mathbb{R}.$$

It is immediately found that $\mathcal{M}_1(z)$ is invertible and, moreover, $\mathcal{M}_1(z) \in \mathbb{T}$ whenever $z \in \mathbb{R}$. For the sake of generality, we also introduce Moebius transformations of the form

$$\mathcal{M}_2(z) = \frac{z-\beta}{1-\bar{\beta}z}, \quad |\beta| \neq 1,$$

which are invertible and map the unit circle $\mathbb{T}$ into itself. Then, by composition of $\mathcal{M}_2(z)$ with $\mathcal{M}_1(z)$ we find a fairly general transformation $\mathcal{M}(z)$ mapping the real axis into the unit circle:

$$\mathcal{M}(z) = \mathcal{M}_2(\mathcal{M}_1(z)) = \frac{(1-\beta)z + (\bar{\alpha}-\beta\alpha)}{(1-\bar{\beta})z + (\alpha-\bar{\alpha}\bar{\beta})}. \tag{4.3}$$

Hence, the inverse transformation $\mathcal{M}^{-1}(z) = \mathcal{M}_1^{-1}(\mathcal{M}_2^{-1}(z))$, where

$$\mathcal{M}_1^{-1}(z) = \frac{\alpha z - \bar{\alpha}}{-z+1}, \quad \mathcal{M}_2^{-1}(z) = \frac{z+\beta}{\bar{\beta}z+1}$$

is the desired invertible transformation which maps the unit circle (except for one point) into the real axis.

By combining these properties with Theorem 4.1, we obtain efficient procedures for the solution of Problem 2 in the case where all the points z_i belong to the unit circle $\mathbb{T}$.

Let $D_y = \operatorname{diag}[y_0, \ldots, y_n]$ and $D_z = \operatorname{diag}[z_0, \ldots, z_n]$ with $|z_i| = 1$. Moreover, let $\mathcal{M}(z)$ be as in (4.3), such that $\mathcal{M}^{-1}(z_i)$ and $\mathcal{M}^{-1}(y_i)$ are finite, i.e., $z_i, y_i \neq (1-\beta)/(1-\bar{\beta}) = \mathcal{M}_2(1)$, $0 \leq i \leq n$. The solution (Q, S) of Problem 2 with input data $\vec{w}$, $\{\mathcal{M}^{-1}(z_i)\}$, and $\{\mathcal{M}^{-1}(y_i)\}$ is such that

$$Q^H \operatorname{diag}[\mathcal{M}^{-1}(z_0), \ldots, \mathcal{M}^{-1}(z_n)]Q = S + \operatorname{diag}[\mathcal{M}^{-1}(y_0), \ldots, \mathcal{M}^{-1}(y_n)],$$

from which it follows that

$$\mathcal{M}(Q^H \operatorname{diag}[\mathcal{M}^{-1}(z_0), \ldots, \mathcal{M}^{-1}(z_n)]Q) = \mathcal{M}(S + \operatorname{diag}[\mathcal{M}^{-1}(y_0), \ldots, \mathcal{M}^{-1}(y_n)]).$$

By invoking Theorem 4.1, this relation gives

$$\mathcal{M}(Q^H \operatorname{diag}[\mathcal{M}^{-1}(z_0), \ldots, \mathcal{M}^{-1}(z_n)]Q) = Q^H D_z Q = \widehat{S} + D_y, \quad \widehat{S} \in \mathcal{S}_{n+1},$$

and, therefore, a solution of the original inverse eigenvalue problem with points $z_i \in \mathbb{T}$ is $(\widehat{Q}, \widehat{S})$, where $\widehat{Q} = Q$ and $\widehat{S}$ is such that

$$\widehat{S} + D_y = \mathcal{M}(S + \mathrm{diag}[\mathcal{M}^{-1}(y_0), \ldots, \mathcal{M}^{-1}(y_n)]). \tag{4.4}$$

Having shown in (4.1) that the matrix S satisfies

$$S = \mathrm{tril}(\vec{u}\vec{v}^T, 0) + \mathrm{triu}(\bar{\vec{v}}\vec{u}^H, 1),$$

for suitable vectors $\vec{u}$ and $\vec{v}$, we can use (4.4) to further investigate the structure of $\widehat{S}$. From (4.4) we deduce that

$$\widehat{S}^H + D_y^H = \tilde{\mathcal{M}}(S^H + \mathrm{diag}[\mathcal{M}^{-1}(y_0), \ldots, \mathcal{M}^{-1}(y_n)]^H).$$

The Moebius transformation $\tilde{\mathcal{M}}$ of a matrix S is defined as

$$\tilde{\mathcal{M}} = (\bar{\gamma}S + \bar{\delta}I)^{-1}(\bar{\alpha}S + \bar{\beta}I)$$

when $\mathcal{M} = (\alpha z + \beta)/(\gamma z + \delta)$. By applying Theorem 4.1 again, assuming that all y_i are different from zero, this implies that

$$\widehat{S}^H + D \in \mathcal{S}_{n+1}$$

for a certain diagonal matrix D. Summing up, we obtain that

$$\widehat{S} = \mathrm{tril}(\vec{u}\vec{v}^T, 0) + \mathrm{triu}(\vec{p}\vec{q}^T, 1), \tag{4.5}$$

for suitable vectors $\vec{u}$, $\vec{v}$, $\vec{p}$, and $\vec{q}$. In case one or more of the y_i are equal to zero, it can be shown that $\widehat{S}$ is block lower triangular where each of the diagonal blocks has the desired structure. The proof is rather technical. Therefore, we omit it here.

From a computational viewpoint, these results can be used to devise several different procedures for solving Problem 2 in the case of points z_i lying on the unit circle at the cost of $O(n^2)$ ops. By taking into account the semiseparable structure of $\widehat{S}$ (4.5) we can simply modify the algorithm stated in the previous section in such a way to compute its upper triangular part without performing multiplications (3.4). A different approach is outlined in the next subsection.

4.3. Special case: All points z_i lie on a generic circle. Another approach to deal with the preceding special case, which immediately generalizes to the case where the nodes z_i belong to a given circle in the complex plane-like $\{z \in \mathbb{C} : |z - p| = r\}$, exploits an invariance property of Cauchy-like matrices under a Moebius transformation of the nodes. Such property is presented in the next lemma for the case of classical Cauchy matrices; the Cauchy case can be dealt with by introducing suitable diagonal scalings. With minor changes, all forthcoming arguments also apply to the case where all abscissas lie on a generic line in the complex plane, since the image of $\mathbb{R}$ under a Moebius transformation is either a circle or a line.

LEMMA 4.2. *Let z_i, y_j, for $1 \le i, j \le n$, be pairwise distinct complex numbers, let*

$$\mathcal{M}(z) = \frac{\alpha z + \beta}{\gamma z + \delta}, \quad \alpha\delta - \beta\gamma \neq 0,$$

be a Moebius transformation, and let $C_{\mathcal{M}} \equiv (1/(\mathcal{M}(z_i) - \mathcal{M}(y_j)))$. Then $C_{\mathcal{M}}$ is a Cauchy-like matrix with nodes z_i, y_j.

Proof. Using the notation above, we have

$$\frac{1}{\mathcal{M}(z_i)-\mathcal{M}(y_j)}=\frac{1}{\alpha\delta-\beta\gamma}\frac{(\gamma z_i+\delta)(\gamma y_j+\delta)}{z_i-y_j}.$$

Hence $C_{\mathcal{M}}$ has the form $C_{\mathcal{M}}\equiv(u_iv_j/(z_i-y_j))$. ∎

In the next theorem, we show how to construct a Moebius transformation mapping $\mathbb{R}$ onto a prescribed circle without one point, thus generalizing formula (4.3). Together with the preceding lemma, it will allow us to translate Problem 2 with nodes on a circle into a corresponding problem with real nodes. The latter can be solved with the technique laid down in subsection 4.1.

THEOREM 4.3. *Let the center of the circle $p\in\mathbb{C}$ and its radius $r>0$ be given. Consider the following algorithm:*

1. *Choose arbitrary nonzero complex numbers $\gamma=|\gamma|\mathrm{e}^{\mathrm{i}\theta_\gamma}$ and $\delta=|\delta|\mathrm{e}^{\mathrm{i}\theta_\delta}$ such that $\mathrm{e}^{2\mathrm{i}(\theta_\gamma-\theta_\delta)}\neq 1$; moreover, choose $\tilde\theta\in[0,2\pi]$.*
2. *Set $\alpha=p\gamma+r|\gamma|\mathrm{e}^{\mathrm{i}\tilde\theta}$.*
3. *Set $\hat\theta=\tilde\theta+\theta_\gamma-\theta_\delta$.*
4. *Set $\beta=p\delta+r|\delta|\mathrm{e}^{\mathrm{i}\hat\theta}$.*

Then the function $\mathcal{M}(z)=(\alpha z+\beta)/(\gamma z+\delta)$ is a Moebius transformation mapping the real line onto the circle $\{z\in\mathbb{C}:|z-p|=r\}$ without the point $\hat z=\alpha/\gamma$.

Proof. After simple manipulations,

$$\left|\frac{\alpha z+\beta}{\gamma z+\delta}-p\right|^2=r^2$$

leads to

$$\begin{aligned}&z^2|\alpha-p\gamma|^2+2z\Re((\alpha-p\gamma)\overline{(\beta-p\delta)})+|\beta-p\delta|^2\\&\quad=z^2r^2|\gamma|^2+2zr^2\Re(\gamma\bar\delta)+r^2|\delta|^2.\end{aligned}\tag{4.6}$$

Here and in what follows, $\Re(z)$ denotes the real part of $z\in\mathbb{C}$. By construction, we have $|\alpha-p\gamma|=r|\gamma|$ and $|\beta-p\delta|=r|\delta|$. Moreover,

$$\begin{aligned}\Re((\alpha-p\gamma)\overline{(\beta-p\delta)})&=r^2|\gamma\delta|\Re(\mathrm{e}^{\mathrm{i}(\tilde\theta-\hat\theta)})\\&=r^2|\gamma\delta|\Re(\mathrm{e}^{\mathrm{i}(\theta_\delta-\theta_\gamma)})\\&=r^2\Re(\gamma\bar\delta).\end{aligned}$$

Hence (4.6) is fulfilled for any real z. The missing point is given by

$$\hat z=\lim_{z\to\infty}\frac{\alpha z+\beta}{\gamma z+\delta}=\frac{\alpha}{\gamma}.$$

It remains for us to prove that $\alpha\delta-\beta\gamma\neq 0$. Indeed, we have

$$\begin{aligned}\alpha\delta-\beta\gamma&=(p\gamma+r|\gamma|\mathrm{e}^{\mathrm{i}\tilde\theta})\delta-(p\delta+r|\delta|\mathrm{e}^{\mathrm{i}\hat\theta})\gamma\\&=r|\gamma|\delta\mathrm{e}^{\mathrm{i}\tilde\theta}-r\gamma|\delta|\mathrm{e}^{\mathrm{i}\hat\theta}\\&=r|\gamma\delta|(\mathrm{e}^{\mathrm{i}(\tilde\theta+\theta_\delta)}-\mathrm{e}^{\mathrm{i}(\hat\theta+\theta_\gamma)})\\&=r|\gamma\delta|\mathrm{e}^{\mathrm{i}(\tilde\theta+\theta_\delta)}(1-\mathrm{e}^{2\mathrm{i}(\theta_\gamma-\theta_\delta)}).\end{aligned}$$

Since $\mathrm{e}^{2\mathrm{i}(\theta_\gamma+\theta_\delta)}\neq 1$ we obtain $\alpha\delta-\beta\gamma\neq 0$. ∎

Suppose we want to solve Problem 2 with data w_i, z_i, y_i, where $|z_i - p| = r$. As seen from the proof of Theorem 2.2, if we let $C \equiv (w_{i-1}(z_{i-1} - y_0)/(z_{i-1} - y_{j-1}))$ and $C = QR$, then a solution is (Q, S), with $S = Q^H D_z Q - D_y$. Let $\mathcal{M}(z) = (\alpha z + \beta)/(\gamma z + \delta)$ be a Moebius transformation built from Theorem 4.3. Recalling the inversion formula (4.2), let $\tilde{z}_i = \mathcal{M}^{-1}(z_i)$, $\tilde{y}_i = \mathcal{M}^{-1}(y_i)$, $v_i = \gamma \tilde{y}_i + \delta$, and

$$\tilde{w}_i = w_i \frac{z_i - y_0}{\tilde{z}_i - \tilde{y}_0} \frac{\gamma \tilde{z}_i + \delta}{\alpha\delta - \beta\gamma}, \qquad 0 \le i \le n.$$

Note that $\tilde{z}_i \in \mathbb{R}$, by construction. From Lemma 4.2, we also have

$$C \equiv \left(\frac{\tilde{w}_{i-1}(\tilde{z}_{i-1} - \tilde{y}_0) v_{j-1}}{\tilde{z}_{i-1} - \tilde{y}_{j-1}} \right).$$

Again from Theorem 2.2, we see that the solution of Problem 2 with data $\tilde{w}_i, \tilde{z}_i, \tilde{y}_i$ is $(Q, \widetilde{S})$, where

$$\widetilde{S} = Q^H \mathcal{M}^{-1}(D_z) Q - \mathcal{M}^{-1}(D_y).$$

Let $\widehat{S} = \widetilde{S} + \mathcal{M}^{-1}(D_y)$. Observe that $\widehat{S}$ is a *diagonal-plus-semiseparable matrix* [12, 14, 18]. After simple passages, we have

$$S = \mathcal{M}(\widehat{S}) - D_y = [\alpha \widehat{S} + \beta I][\gamma \widehat{S} + \delta I]^{-1} - D_y.$$

Hence S can be recovered from $\widetilde{S}$ by determining the entries in its first and last rows and columns. This latter task can be carried out at a linear cost by means of several different algorithms for the solution of diagonal-plus-semiseparable linear systems; see, e.g., [12, 14, 26, 38].

5. Computational issues. In this section we discuss some numerical and computational issues concerned with the application of the algorithm stated in section 3 for the solution of the inverse eigenvalue problem 2 (DS-IEP). We implemented the $\mathcal{O}(n^2)$ algorithm in Matlab and performed extensive numerical experiments in order to compare the numerical behavior of our algorithm with other existing $\mathcal{O}(n^2)$ solution methods based on the computation of the columns of the matrix Q by means of a three-term recurrence relation [16, 17]. We have tried a number of different data sets and the algorithm which we developed here always returned better results than the algorithms of [16, 17]. For instance, consider the following data set: $w_i = 1$, $z_i = i + n$ and $y_i = i + n - \frac{1}{2}$ for $i = 0, 1, 2, \ldots, n$. To show that our algorithm is indeed $\mathcal{O}(n^2)$, we plot in Figure 5.1 the execution time divided by n^2 for the different sizes of the problem. Here we set $n = 10, 20, 30, \ldots, 1000$. The slight deviation of the graph from a straight horizontal line can be attributed to the memory management overhead, due to computations with larger matrices.

Figure 5.2 gives the maximum relative error on the eigenvalues of the computed diagonal-plus-semiseparable matrix compared to the original points z_i for $n = 10, 20, 30, \ldots, 500$. In Figure 5.3, the same is done for the weights. Figures 5.2 and 5.3 show that the algorithm is accurate for this specific data set. Differently, the algorithms of [16, 17] already provide inaccurate results for small values of n, say $n > 30$, due to the loss of orthogonality of the columns of Q.

However, it is worth mentioning that in our experience we found examples for which our algorithm does not perform so well as in the previous case. Experimentally, it was noticed that much depends on the balancing of the vectors $\vec{u}_k$ and $\vec{v}_k$ which

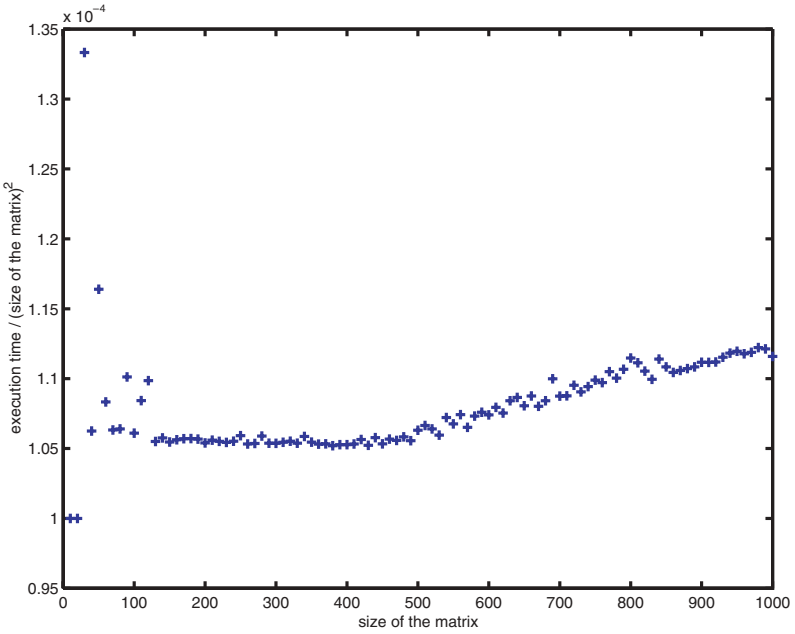


FIG. 5.1. *Computational complexity.*

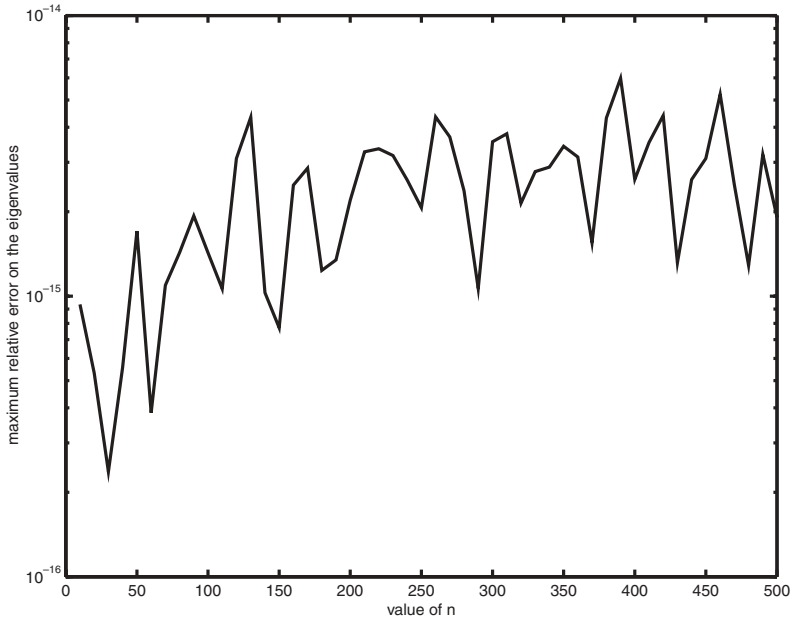


FIG. 5.2. *Relative accuracy of the eigenvalues.*

define the lower triangular part of the matrix $S_k \in \mathcal{S}_{k+1}$, recursively generated by our algorithm to compute the final matrix $S = S_n \in \mathcal{S}_{n+1}$. The computation of an unbalanced representation is an ill-conditioned task resulting in less accurate results.

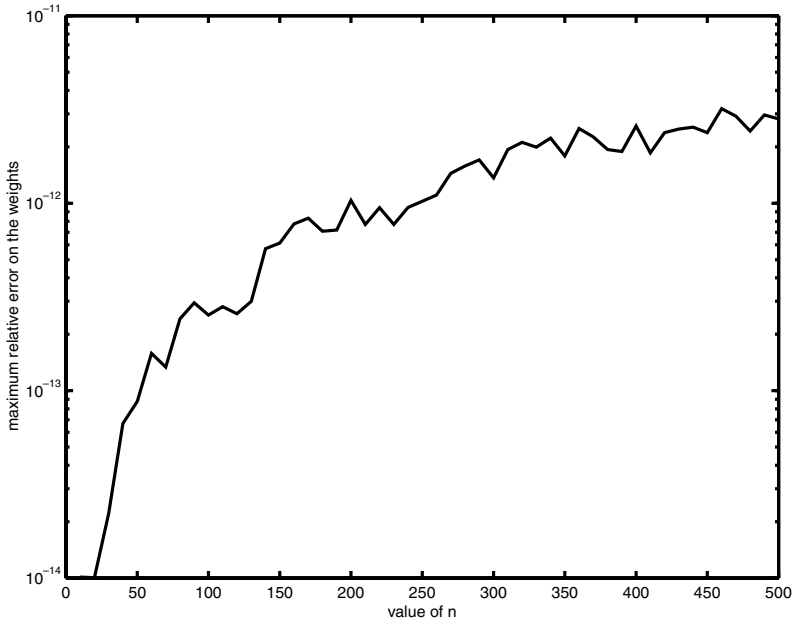


FIG. 5.3. *Relative accuracy of the weights.*

To see this, let us consider the matrix

$$S = \begin{bmatrix} 1 & * \\ \epsilon & 1 \end{bmatrix},$$

where $*$ denotes some number and $\epsilon > 0$ is small. The lower triangular part of this matrix admits a representation by means of the two vectors $\vec{u} = [1, \epsilon]$ and $\vec{v} = [1, 1/\epsilon]$. Suppose now we perturb the matrix S by adding a perturbation matrix ΔS, $\|\Delta S\| = \mathcal{O}(\epsilon)$, such that

$$S + \Delta S = \begin{bmatrix} 1 & * \\ \eta & 1 \end{bmatrix},$$

where $0 < \eta < \epsilon$. Then, the lower triangular part of the perturbed matrix $S + \Delta S$ can be expressed by using the two vectors $\vec{u} = [1, \eta]$ and $\vec{v} = [1, 1/\eta]$. This means that a perturbation of order ϵ of the matrix S produces a vector $\vec{v}$ whose second component is affected by an absolute error equal to $|1/\epsilon - 1/\eta| = \mathcal{O}(1/\eta)$.

Similar numerical problems were also encountered in the solution of the direct eigenvalue problem for semiseparable matrices by means of QR iteration [4, 39]. To circumvent these difficulties, in the cited papers novel numerically robust representations of semiseparable matrices have been introduced and analyzed. They look like

$$S = \begin{bmatrix} u_1 v_1 & * & \cdots & * \\ u_2 t_1 v_1 & u_2 v_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & * \\ u_n t_{n-1} \dots t_1 v_1 & u_n t_{n-1} \dots t_2 v_2 & \cdots & u_n v_n \end{bmatrix}.$$

If we make use of such a representation for the 2×2 matrix $S \in \mathcal{S}_2$ described above, then it is easy to verify that the considered $\mathcal{O}(\epsilon)$ perturbation only causes $\mathcal{O}(\epsilon)$ changes in the elements of the representation.

The adaptation of the algorithm presented in this paper to deal with such modified representations is an ongoing work and, in our opinion, will lead to the design of a stable algorithm for solving the inverse eigenvalue problems which exploits the computational properties of semiseparable matrices.

REFERENCES

[1] G. Ammar and W. Gragg, *$O(n^2)$ reduction algorithms for the construction of a band matrix from spectral data*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 426–431.

[2] G. Ammar, W. Gragg, and L. Reichel, *Constructing a unitary Hessenberg matrix from spectral data*, in Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms, G. Golub and P. Van Dooren, eds., NATO Adv. Sci. Inst. Ser. F: Comput. Systems Sci. 70, Springer-Verlag, Berlin, 1991, pp. 385–395.

[3] E. Asplund, *Inverses of matrices a_i, j which satisfy $a_{i,j} = 0$ for $j > i + p$*, Math. Scand., 7 (1959), pp. 57–60.

[4] D. A. Bini, L. Gemignani, and V. Y. Pan, *QR-like Algorithms for Generalized Semiseparable Matrices*, Technical Report 1470, Dipartimento di Matematica, Università di Pisa, Pisa, Italy, 2003.

[5] D. Boley and G. Golub, *A survey of matrix inverse eigenvalue problems*, Inverse Problems, 3 (1987), pp. 595–622.

[6] A. Bultheel and M. Van Barel, *Padé techniques for model reduction in linear system theory: A survey*, J. Comput. Appl. Math., 14 (1986), pp. 401–438.

[7] A. Bultheel and B. De Moor, *Rational approximation in linear systems and control*, J. Comput. Appl. Math., 121 (2000), pp. 355–378.

[8] A. Bultheel, P. González-Vera, E. Hendriksen, and O. Njåstad, *Orthogonal rational functions with poles on the unit circle*, J. Math. Anal. Appl., 182 (1994), pp. 221–243.

[9] A. Bultheel, P. González-Vera, E. Hendriksen, and O. Njåstad, *Orthogonal Rational Functions*, Cambridge Monogr. Appl. Comput. Math. 5, Cambridge University Press, Cambridge, UK, 1999.

[10] A. Bultheel, P. González-Vera, E. Hendriksen, and O. Njåstad, *Quadrature and orthogonal rational functions*, J. Comput. Appl. Math., 127 (2001), pp. 67–91.

[11] A. Bultheel and M. Van Barel, *Vector orthogonal polynomials and least squares approximation*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 863–885.

[12] S. Chandrasekaran and M. Gu, *A fast and stable solver for recursively semi-separable systems of linear equations*, in Structured Matrices in Mathematics, Computer Science, and Engineering, II (Boulder, CO, 1999), Contemp. Math. 281, AMS, Providence, RI, 2001, pp. 39–53.

[13] P. Delsarte, Y. Genin, and Y. Kamp, *On the role of the Nevanlinna-Pick problem in circuit and system theory*, Internat. J. Circuit Theory Appl., 9 (1981), pp. 177–187.

[14] Y. Eidelman and I. Gohberg, *A look-ahead block Schur algorithm for diagonal plus semiseparable matrices*, Comput. Math. Appl., 35 (1998), pp. 25–34.

[15] S. Elhay, G. Golub, and J. Kautsky, *Updating and downdating of orthogonal polynomials with data fitting applications*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 327–353.

[16] D. Fasino and L. Gemignani, *A Lanczos-type algorithm for the QR factorization of regular Cauchy matrices*, Numer. Linear Algebra Appl., 9 (2002), pp. 305–319.

[17] D. Fasino and L. Gemignani, *A Lanczos-type algorithm for the QR factorization of Cauchy-like matrices*, in Fast Algorithms for Structured Matrices: Theory and Applications, Contemp. Math. 323, V. Olshevsky, ed., AMS, Providence, RI, 2003, pp. 91–104.

[18] D. Fasino and L. Gemignani, *Direct and inverse eigenvalue problems for diagonal-plus-semiseparable matrices*, Numer. Algorithms, 34 (2003), pp. 313–324.

[19] B. Fischer and G. Golub, *How to generate unknown orthogonal polynomials out of known orthogonal polynomials*, J. Comput. Appl. Math., 43 (1992), pp. 99–115.

[20] F. R. Gantmacher and M. G. Krein, *Sur les matrices complètement non négatives et oscillatoires*, Compositio Math., 4 (1937), pp. 445–470.

[21] W. Gautschi, *The use of rational functions in numerical quadrature*, J. Comput. Appl. Math., 133 (2001), pp. 111–126.

[22] G. H. Golub and J. H. Welsch, *Calculation of Gauss quadrature rules*, Math. Comp., 23

(1969), pp. 221–230.
[23] W. Gragg and A. Lindquist, *On the partial realization problem*, Linear Algebra Appl., 50 (1983), pp. 277–319.
[24] W. B. Gragg and W. J. Harrod, *The numerically stable reconstruction of Jacobi matrices from spectral data*, Numer. Math., 44 (1984), pp. 317–335.
[25] P. Henrici, *Applied and Computational Complex Analysis*, Vol. 1, John Wiley, New York, 1974.
[26] I. Koltracht, *Linear complexity algorithm for semiseparable matrices*, Integral Equations Operator Theory, 29 (1997), pp. 313–319.
[27] B. Ninness and F. Gustafsson, *A unifying construction of orthonormal bases for system identification*, IEEE Trans. Automat. Control, 42 (1997), pp. 515–521.
[28] V. Olshevsky and V. Pan, *Polynomial and rational evaluation and interpolation (with structured matrices)*, in Automata, Languages and Programming, Lecture Notes in Comput. Sci. 1644, Springer-Verag, Berlin, 1999, pp. 585–594.
[29] L. Reichel, *Fast QR decomposition of Vandermonde-like matrices and polynomial least squares approximation*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 552–564.
[30] L. Reichel, *Construction of polynomials that are orthogonal with respect to a discrete bilinear form*, Adv. Comput. Math., 1 (1993), pp. 241–258.
[31] L. Reichel, G. Ammar, and W. Gragg, *Discrete least squares approximation by trigonometric polynomials*, Math. Comp., 57 (1991), pp. 273–289.
[32] P. Rózsa, *On the inverses of band matrices*, Integral Equations Operator Theory, 10 (1987), pp. 82–95.
[33] H. Rutishauser, *On Jacobi rotation patterns*, in Experimental Arithmetic, High Speed Computing and Mathematics, Proc. Sympos. Appl. Math. 15, AMS, Providence, RI, 1963, pp. 219–239.
[34] M. Van Barel and A. Bultheel, *A new approach to the rational interpolation problem: The vector case.*, J. Comput. Appl. Math., 33 (1990), pp. 331–346.
[35] M. Van Barel and A. Bultheel, *A parallel algorithm for discrete least squares rational approximation*, Numer. Math., 63 (1992), pp. 99–121.
[36] M. Van Barel and A. Bultheel, *Discrete linearized least-squares rational approximation on the unit circle*, J. Comput. Appl. Math., 50 (1994), pp. 545–563.
[37] M. Van Barel and A. Bultheel, *Orthonormal polynomial vectors and least squares approximation for a discrete inner product*, Electron. Trans. Numer. Anal., 3 (1995), pp. 1–23.
[38] E. Van Camp, N. Mastronardi, and M. Van Barel, *Two fast algorithms for solving diagonal-plus-semiseparable linear systems*, J. Comput. Appl. Math., 164–165 (2004), pp. 731–747.
[39] R. Vandebril, M. Van Barel, and N. Mastronardi, *An implicit QR algorithm for semiseparable matrices to compute the eigendecomposition of symmetric matrices*, Numer. Linear Algebra Appl., to appear.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 3, pp. 830–848

ON STABLE EIGENDECOMPOSITIONS OF MATRICES*

R. ALAM† AND S. BORA†

Abstract. A general framework is presented for analyzing the continuous evolution of eigendecompositions of matrices. More specifically, for arbitrary operator norms, a general framework based on pseudospectra of matrices is developed for computing upper bounds on ϵ from the Schur or block Schur forms of a complex n-by-n matrix A that ensure stability of eigendecompositions of A when A varies in the ball $\{A' : \|A - A'\| \leq \epsilon\}$. For the 2-norm and the Frobenius norm, the new bounds presented compare well with the bounds obtained by Demmel and Wilkinson.

Key words. eigendecomposition, ϵ-pseudospectrum, geometric separation

AMS subject classifications. 65F15, 15A18, 15A12, 65F35, 65F30

DOI. 10.1137/S0895479802418458

1. Introduction.

Computation of eigendecompositions of matrices is an important task in numerical linear algebra. By an eigendecomposition of a matrix $A \in \mathbb{C}^{n\times n}$, we mean a decomposition of A of the form

$$A = X\mathrm{diag}(A_1, \ldots, A_m)X^{-1}, \text{ where } \sigma(A_i) \cap \sigma(A_j) = \emptyset \text{ for } i \neq j. \tag{1.1}$$

Here $\sigma(A_j)$ is the spectrum of A_j. Following [4], we also specify an eigendecomposition of A by a disjoint partition of the spectrum $\sigma(A)$:

$$\sigma(A) = \cup_{j=1}^{m} \sigma_j. \tag{1.2}$$

Clearly, the sensitivity of an eigendecomposition of A is strongly influenced by the partition of $\sigma(A)$ induced by the eigendecomposition. For example, in an eigendecomposition of A if two close eigenvalues are made to appear in two disjoint parts of $\sigma(A)$, then the eigendecomposition tends to be highly sensitive to perturbation. We mention that finding a well-conditioned eigendecomposition of A is a hard problem. It is shown in [9] that the problem of finding a well-conditioned block diagonalizing similarity transformation is in general NP-hard. More specifically, given a tolerance τ, the problem of finding an eigendecomposition of A of the form (1.1) such that the condition number of X is at most τ is NP-hard.

Fortunately, most often the task is to compute an eigendecomposition specified by a specific application. There are robust algorithms to compute a specified eigendecomposition of a matrix which also attempt to control the condition number of the block diagonalizing similarity transformation in terms of input tolerance (see, for example, [2], [11]). Since finite precision affects the computation, it is important to understand the effect of small perturbations on an eigendecomposition of A. The effect of small perturbation on an eigendecomposition of A has been analyzed, for example, in [4], [22], [23], [16], [20].

The main objective of this paper is to analyze evolution of eigendecompositions. Given an eigendecomposition of A of the form (1.2), we analyze its evolution in a

*Received by the editors November 23, 2002; accepted for publication (in revised form) by B. T. Kågström July 1, 2004; published electronically April 8, 2005.
http://www.siam.org/journals/simax/26-3/41845.html

†Department of Mathematics, Indian Institute of Technology Guwahati, Guwahati, 781 039, India (rafik@iitg.ernet.in, rafikul@yahoo.com, shreemayee@yahoo.com). The research of the second author was funded by the CSIR, Government of India.

small neighborhood of A. More specifically, we analyze the continuous evolution of (1.2) on the closed ball

$$\mathbf{A}(\epsilon) := \{A' \in \mathbb{C}^{n\times n} : \|A - A'\| \le \epsilon\}.$$

By continuous evolution of (1.2) on $\mathbf{A}(\epsilon)$, we mean that for each $j = 1, 2, \ldots, m$, the spectral projection P_j associated with A and σ_j varies continuously when A varies in $\mathbf{A}(\epsilon)$. For example, if (1.2) evolves continuously on $\mathbf{A}(\epsilon)$, then for each $\|E\| \le \epsilon$, $A+E$ admits an eigendecomposition $\sigma(A+E) = \cup_{j=1}^{m}\sigma_j'$ such that each σ_j' results from the *continuous deformation* of σ_j.

The ϵ-pseudospectra, well known for analyzing behavior of nonnormal matrices [17], [18], arise naturally for analyzing stability of eigendecompositions. As shown in [1], [3], the ϵ-pseudospectrum $\Lambda_\epsilon(A)$ (defined in section 2) provides a natural framework for analyzing the continuous evolution of eigendecompositions of A on $\mathbf{A}(\epsilon)$. In fact, for the 2-norm and, under appropriate assumption, the Frobenius norm, a necessary and sufficient condition for continuous evolution of (1.2) on $\mathbf{A}(\epsilon)$ can be read off from $\Lambda_\epsilon(A)$. Since computation of pseudospectra is expensive, for practical purposes, it is desirable to have upper bounds of ϵ that ensure the continuous evolution of (1.2) on $\mathbf{A}(\epsilon)$. For example, when A is of the form

$$A := \begin{bmatrix} A_1 & C \\ 0 & A_2 \end{bmatrix}, \text{ where } \sigma(A_1) \cap \sigma(A_2) = \emptyset, \tag{1.3}$$

it was shown by Demmel [4] that $\sigma(A) = \sigma(A_1) \cup \sigma(A_2)$ evolves continuously on $\mathbf{A}(\epsilon)$ if

$$\epsilon < \frac{\text{sep}_\lambda(A_1, A_2)}{\|P\|_2 + \sqrt{\|P\|_2^2 - 1}}. \tag{1.4}$$

Here P is the spectral projection associated with A and $\sigma(A_1)$, and sep_λ is the spectral overlap [4], [19] of A_1 and A_2 (defined in section 4). A slightly weaker bound $\epsilon < \text{sep}_\lambda(A_1, A_2)/2\|P\|_2$ was obtained by Wilkinson [22], [23]. A more conservative bound $\epsilon < \text{sep}(A_1, A_2)/4\|P\|_2$ is due to Stewart [16], where $\text{sep}(A_1, A_2)$ is the separation of A_1 and A_2 (defined in section 4).

For arbitrary operator norms, we describe a general framework for obtaining various upper bounds of ϵ that ensure the continuous evolution of eigendecompositions of A on $\mathbf{A}(\epsilon)$. We show that various upper bounds of ϵ are easy consequences of inclusion theorems for $\Lambda_\epsilon(A)$. In fact, we show that if A is similar to $\text{diag}(A_1, A_2)$, where $\sigma(A_1) \cap \sigma(A_2) = \emptyset$, and

$$\Lambda_\epsilon(A) \subset \Lambda_{\phi(\epsilon)}(A_1) \cup \Lambda_{\phi(\epsilon)}(A_2)$$

for some strictly increasing function ϕ, then $\sigma(A) = \sigma(A_1) \cup \sigma(A_2)$ evolves continuously on $\mathbf{A}(\epsilon)$ if $\epsilon < \phi^{-1}(\text{sep}_\lambda(A_1, A_2))$. In particular, when A is given by (1.3), for the spectral and the Frobenius norms we show that $\sigma(A) = \sigma(A_1) \cup \sigma(A_2)$ evolves continuously on $\mathbf{A}(\epsilon)$ if

$$\epsilon < \frac{2\left(\text{sep}_\lambda(A_1, A_2)\right)^2}{\|C\|_2 + \sqrt{\|C\|_2^2 + 4\left(\text{sep}_\lambda(A_1, A_2)\right)^2}} \text{ and } \epsilon < \frac{2\left(\text{sep}_\lambda(A_1, A_2)\right)^2}{\|C\|_F + \sqrt{\|C\|_F^2 + 4\left(\text{sep}_\lambda(A_1, A_2)\right)^2}},$$

respectively. Inequality (1.4) is known to be the best upper bound for ϵ. We show that our upper bound for the 2-norm compares well with (1.4) in the sense that there

are matrices for which our bound is better than (1.4), and vice versa. In fact, we show that they differ from each other at most by a factor of sep_λ.

The geometric separation of eigenvalues gsep (defined in section 2), which can be read off from ϵ-pseudospectra, provides a sufficient condition for continuous evolution of eigendecompositions. For example, (1.2) evolves continuously on $\mathbf{A}(\epsilon)$ if $\epsilon < \min_j \mathrm{gsep}(\sigma_j)$. We show that various bounds discussed above are, in fact, lower bounds of the geometric separation gsep and are obtained by approximating $\Lambda_\epsilon(A)$. Thus from our general framework we show that

$$\phi^{-1}(\mathrm{sep}_\lambda(A_1, A_2)) \le \mathrm{gsep}(\sigma(A_1), \sigma(A_2)).$$

This paper is organized as follows. Section 2 briefly reviews ϵ-pseudospectra and their applications in analyzing the stability of eigendecompositions. Section 3 presents localization theorems, old and new, for the perturbed eigenvalues in the setting of ϵ-pseudospectra. The notions of spectral overlap and separation of matrices have been discussed in section 4. Section 5 derives various lower bounds of gsep. Section 6 is devoted to establishing relationships between various concepts such as geometric separation, spectral overlap, dissociation, and separation of eigenvalues. Finally, section 7 shows that for 2-by-2 matrices, gsep attains some of the lower bounds.

Notation. We denote the set of n-by-n complex matrices by $\mathbb{C}^{n\times n}$. For $A \in \mathbb{C}^{n\times n}$, we denote the spectrum of A by $\sigma(A)$. For $z \in \mathbb{C} \setminus \sigma(A)$, we set $R(A, z) := (A - zI)^{-1}$. An operator norm of A is defined by $\|A\| := \sup_{\|x\|=1} \|Ax\|$. The p-norm on $\mathbb{C}^n$ and the induced operator norm is denoted by $\|\cdot\|_p$. The 2-norm is also referred to as the spectral norm. Throughout this paper, we consider operator norms, the only exception being the Frobenius norm $\|A\|_F := \sqrt{\mathrm{trace}(A^*A)}$. We, however, mention that most results in this paper hold for any submultiplicative norms. We say that $\|\cdot\|$ is a max-norm if $\|\mathrm{diag}(A_1, A_2)\| = \max(\|A_1\|, \|A_2\|)$. For example, each p-norm is a max-norm. If X is invertible, then $K(X) := \|X^{-1}\|\,\|X\|$ is the condition number of X. For the p-norm, $K(X)$ is denoted by $K_p(X)$. Finally, we denote the closed ball in $\mathbb{C}$ by $B[z, r]$, that is, $B[z, r] := \{w \in \mathbb{C} : |w - z| \le r\}$.

2. Stability of eigendecompositions. Let $A \in \mathbb{C}^{n\times n}$. We briefly show that the ϵ-pseudospectrum provides a natural framework for analyzing stability of eigendecompositions. Consider the eigendecomposition (1.1). Partition $X = [X_1, \ldots, X_m]$ and $Y := (X^{-1})^* = [Y_1, \ldots, Y_m]$, the partitioning being conformal with that of $\mathrm{diag}(A_1, \ldots, A_m)$. Then the columns of X_i (resp., Y_i) span the right (resp., left) invariant subspace of A corresponding to $\sigma(A_i)$. Further, $P_i := X_i Y_i^*$ is the spectral projection associated with A and $\sigma(A_i)$, $P_i P_j = 0$ for $i \ne j$, and $P_1 + \cdots + P_m = I$. Recall that $\mathbf{A}(\epsilon) := \{A' : \|A - A'\| \le \epsilon\}$. Following [4], we identity A with $\mathbf{A}(\epsilon)$ and say that A is known to within the tolerance ϵ.

DEFINITION 2.1. *Let $\epsilon > 0$. An eigendecomposition $\sigma(A) = \cup_{j=1}^m \sigma_j$ is said to be ϵ-stable if the spectral projection P_j associated with A and σ_j varies continuously when A varies in $\mathbf{A}(\epsilon)$ for all $j = 1, 2, \ldots, m$.*

Clearly, (1.2) is ϵ-stable if and only if an eigenvalue from σ_i and an eigenvalue from σ_j do not move and coalesce when A varies in $\mathbf{A}(\epsilon)$ for all $i \ne j$. This shows that for analyzing stability of (1.2), we need to understand the evolution of eigenvalues of A when A varies in $\mathbf{A}(\epsilon)$. Consequently, we are led to consider the ϵ-pseudospectrum $\Lambda_\epsilon(A)$ of A given by

$$\Lambda_\epsilon(A) := \bigcup_{A' \in \mathbf{A}(\epsilon)} \sigma(A').$$

For operator norms, $\Lambda_\epsilon(A) = \{z \in \mathbb{C} : \|R(A,z)\| \geq \epsilon^{-1}\}$ (see [17], [18]). We note, however, that for nonoperator norms, these two definitions are not equivalent. The only nonoperator norm we use in this paper is the Frobenius norm. However, as shown in [1], $\Lambda_\epsilon(A)$ is the same for the 2-norm and the Frobenius norm, that is,

$$\Lambda_\epsilon(A) = \bigcup_{\|E\|_2 \leq \epsilon} \sigma(A+E) = \bigcup_{\|E\|_F \leq \epsilon} \sigma(A+E). \tag{2.1}$$

Hence we effectively work with pseudospectra for operator norms. It is well known that, for $\epsilon > 0$, $\Lambda_\epsilon(A)$ consists of nontrivial components (i.e., maximal connected sets) and each component of $\Lambda_\epsilon(A)$ contains at least one eigenvalue in its interior (see [3], [7], [10]). Thus if A has k distinct eigenvalues, then for sufficiently small ϵ, $\Lambda_\epsilon(A)$ consists of exactly k components. As ϵ increases gradually the size of these components increases and coalesces with other components.

DEFINITION 2.2 (see [1]). *The geometric separation of an eigenvalue λ from the rest of $\sigma(A)$, denoted by* $\mathrm{gsep}(\lambda)$, *is the smallest value of ϵ for which a component of $\Lambda_\epsilon(A)$ containing λ coalesces with another component of $\Lambda_\epsilon(A)$.*

If σ_1 is a nonempty subset of $\sigma(A)$, then the geometric separation of σ_1 from the rest of $\sigma(A)$, denoted by $\mathrm{gsep}(\sigma_1)$, *is the smallest value of ϵ for which a component of $\Lambda_\epsilon(A)$ containing an eigenvalue from σ_1 coalesces with a component containing an eigenvalue from $\sigma(A) \setminus \sigma_1$.*

The geometric separation, $\mathrm{gsep}(\sigma_1, \ldots, \sigma_m)$, *of an eigendecomposition $\sigma(A) = \cup_{j=1}^m \sigma_j$ is the smallest value of ϵ for which a component of $\Lambda_\epsilon(A)$ containing an eigenvalue from σ_i coalesces with a component containing an eigenvalue from σ_k for some $i \neq k$.*

Notice that gsep depends on A as well as the norm, which we do not show for notational simplicity. We denote $\mathrm{gsep}(\sigma_1)$ by $\mathrm{gsep}(\sigma_1, A)$ whenever it is necessary to show the dependence on A. It is immediate that gsep is the same for the 2-norm and the Frobenius norm and that

$$\mathrm{gsep}(\sigma_1, \ldots, \sigma_m) = \min_{1 \leq j \leq m} \mathrm{gsep}(\sigma_j).$$

It is evident that if $\epsilon < \mathrm{gsep}(\sigma_j)$, then, as A varies in $\mathbf{A}(\epsilon)$, the eigenvalues generated from σ_j remain disjoint from the eigenvalues generated from σ_i for all $i \neq j$. Hence the spectral projection associated with A and σ_j varies continuously when A varies in $\mathbf{A}(\epsilon)$.

THEOREM 2.3 (see [1]). *Eigendecomposition* (1.2) *is ϵ-stable if $\epsilon < \mathrm{gsep}(\sigma_1, \ldots, \sigma_m)$. Equivalently,* (1.2) *is ϵ-stable if each component of $\Lambda_\epsilon(A)$ contains eigenvalues from exactly one of the sets $\sigma_1, \ldots, \sigma_m$.*

For the 2-norm and, under appropriate assumption, the Frobenius norm, (1.2) *is ϵ-stable if and only if $\epsilon < \mathrm{gsep}(\sigma_1, \ldots, \sigma_m)$.*

Observe that for analyzing ϵ-stability of eigendecompositions of A, it is enough to consider an eigendecomposition of the form $\sigma(A) = \sigma_1 \cup \sigma_2$. Further, given an $\epsilon > 0$, we can choose an eigendecomposition of A that is ϵ-stable by looking at $\Lambda_\epsilon(A)$. Indeed, if $\Lambda_\epsilon(A)$ consists of m components $\Delta_1, \ldots, \Delta_m$, then for $\sigma_j := \sigma(A) \cap \Delta_j$, the eigendecomposition $\sigma(A) = \cup_{j=1}^m \sigma_j$ is obviously ϵ-stable.

Although, gsep can be read off from $\Lambda_\epsilon(A)$, the computation of $\Lambda_\epsilon(A)$ is expensive. Therefore it is desirable to have a sharp lower bound of gsep that can be computed from a Schur or a block Schur form of A. In what follows, we describe a general

framework for obtaining various lower bounds of gsep. We show that these bounds are, in fact, easy consequences of approximation results of $\Lambda_\epsilon(A)$.

3. Inclusion theorems for pseudospectra. The ϵ-pseudospectrum $\Lambda_\epsilon(A)$ provides the best possible localization for $\sigma(A+E)$ when $\|E\| \le \epsilon$. Further, a component of $\Lambda_\epsilon(A)$ provides the best possible localization for the perturbed eigenvalues generated from the eigenvalues inside it. There are localization theorems such as the Gershgorin disk theorem, the Bauer–Fike theorem [15], and a block version of it known as the Wilkinson–Feingold theorem [23] which localize perturbed eigen-values in the complex plane. We present some of these theorems in the setting of ϵ-pseudospectra and obtain new inclusion results for $\Lambda_\epsilon(A)$ from which lower bounds of gsep follow. Recall that $B[z,r]$ denotes the closed disk of radius r centered at z, that is, $B[z,r] := \{w \in \mathbb{C} : |z-w| \le r\}$.

THEOREM 3.1 (Bauer–Fike theorem). *Let X be invertible and $\kappa := K(X)$. Then*

$$\Lambda_\epsilon(A) \subset \Lambda_{\kappa\epsilon}(X^{-1}AX) \quad \textit{and} \quad \Lambda_\epsilon(X^{-1}AX) \subset \Lambda_{\kappa\epsilon}(A).$$

Proof. Set $\mathrm{sep}(z,A) := \min\{\|E\| : z \in \sigma(A+E)\}$. Then it is easy to see that $\Lambda_\epsilon(A) = \{z : \mathrm{sep}(z,A) \le \epsilon\}$. Now, the result follows from the fact that $\mathrm{sep}(z,\, X^{-1}AX) \le K(X)\mathrm{sep}(z,A)$. ∎

When A is diagonalizable, the following inclusion results hold for $\Lambda_\epsilon(A)$.

THEOREM 3.2 (Bauer–Fike–Wilkinson theorem [22]). *Suppose that A is diagonalizable and $\sigma(A) = \{\lambda_1, \ldots, \lambda_m\}$. Let P_j be the spectral projection associated with A and λ_j for $j = 1, 2, \ldots, m$.*

(a) *Set $\kappa := \inf\{K(X) : X^{-1}AX = \mathrm{diag}(\lambda_i)\}$. Then for a max-norm, we have*

$$\Lambda_\epsilon(A) \subset \cup_{j=1}^m B[\lambda_j,\ \kappa\,\epsilon].$$

(b) *For an operator norm, $\Lambda_\epsilon(A) \subset \cup_{j=1}^m B[\lambda_j,\ s\epsilon]$, where $s := \sum_{j=1}^m \|P_j\|$.*

(c) *Also, for an operator norm, $\Lambda_\epsilon(A) \subset \cup_{j=1}^m B[\lambda_j,\ m\|P_j\|\,\epsilon]$.*

Proof. Note that (a) follows from Theorem 3.1. Since A is diagonalizable, we have

$$R(A,z) = \frac{P_1}{\lambda_1 - z} + \cdots + \frac{P_m}{\lambda_m - z}.$$

Therefore $\|R(A,z)\| \le \dfrac{\sum_{j=1}^m \|P_j\|}{\min_{1\le j\le m}|z-\lambda_j|}$ and $\|R(A,z)\| \le m \max_{1\le j\le m}\left(\frac{\|P_j\|}{|z-\lambda_j|}\right)$. Hence the results follow. ∎

Clearly the approximation of $\Lambda_\epsilon(A)$ provided by Theorem 3.2 is strongly influenced by the most sensitive eigenvalues of A. A block version of the above results due to Wilkinson and Feingold [23] provides a better approximation of $\Lambda_\epsilon(A)$ when sensitive eigenvalues are grouped into clusters. Consider the eigendecomposition (1.1). If P_j is the spectral projection associated with A and $\sigma(A_j)$, then (see [22], [23])

$$\inf_X\{K_2(X) : X^{-1}AX = \mathrm{diag}(A_1, \ldots, A_m)\} \le \|P_1\|_2 + \cdots + \|P_m\|_2.$$

For a similar bound, see [5]. By Theorem 3.1, we have the following inclusions.

THEOREM 3.3 (Wilkinson–Feingold theorem [22], [23]). *Let* $s := \|P_1\|_2 + \cdots + \|P_m\|_2$ *and* $\epsilon_j := m\|P_j\|_2\,\epsilon$. *Then for the* 2*-norm, and hence for the Frobenius norm, we have*

$$\Lambda_\epsilon(A) \subset \cup_{j=1}^m \Lambda_{s\epsilon}(A_j) \quad \textit{and} \quad \Lambda_\epsilon(A) \subset \cup_{j=1}^m \Lambda_{\epsilon_j}(A_j).$$

More generally, if $\kappa := \inf\{K(X) : X^{-1}AX = \operatorname{diag}(A_1, \ldots, A_m)\}$, *then for a max-norm, we have* $\Lambda_\epsilon(A) \subset \cup_{j=1}^m \Lambda_{\kappa\epsilon}(A_j)$.

Evidently, the approximation of $\Lambda_\epsilon(A)$ provided by Theorem 3.3 is strongly influenced by the most sensitive blocks in $\operatorname{diag}(A_1, \ldots, A_m)$. In other words, the approximation of $\Lambda_\epsilon(A)$ provided by Theorem 3.3 is strongly influenced by the ill-conditioning of the eigendecomposition (1.1).

Next, we obtain inclusion domains for $\Lambda_\epsilon(A)$ when A is block upper triangular. For the rest of this section, we assume that A is given by (1.3). Let P be the spectral projection associated with A and $\sigma(A_1)$. If X is such that $A_1X - XA_2 = C$, then [5], [4]

$$\inf_S\{K_2(S) : S^{-1}AS = \operatorname{diag}(A_1, A_2)\} = \|P\|_2 + \sqrt{\|P\|_2^2 - 1} = \|X\|_2 + \sqrt{\|X\|_2^2 + 1}.$$

Therefore, by Theorem 3.1, we have the following inclusion.

THEOREM 3.4 (Demmel [4]). *Let* $\epsilon_0 := (\|P\|_2 + \sqrt{\|P\|_2^2 - 1})\epsilon$. *For the* 2*-norm,*

$$\Lambda_\epsilon(A) \subset \Lambda_{\epsilon_0}(A_1) \cup \Lambda_{\epsilon_0}(A_2).$$

A weaker inclusion $\Lambda_\epsilon(A) \subset \Lambda_{2\|P\|_2\epsilon}(A_1) \cup \Lambda_{2\|P\|_2\epsilon}(A_2)$ follows from Theorem 3.3. For a max-norm, we have the following approximation of $\Lambda_\epsilon(A)$.

THEOREM 3.5. *Let* X *be the solution of the Sylvester equation* $A_1X - XA_2 = C$. *Let* $g(\epsilon) := (2\|X\| + 1)\epsilon$ *and* $f(\epsilon) := \frac{\epsilon + \sqrt{\epsilon^2 + 4\,\|C\|\,\epsilon}}{2}$. *Then for a max-norm,*

(a) $\Lambda_\epsilon(A) \subset \Lambda_{g(\epsilon)}(A_1) \cup \Lambda_{g(\epsilon)}(A_2)$,
(b) $\Lambda_\epsilon(A) \subset \Lambda_{f(\epsilon)}(A_1) \cup \Lambda_{f(\epsilon)}(A_2)$.

Proof. Set $r := \|X\|$ and $c := \|C\|$. For $z \in \mathbb{C} \setminus \sigma(A)$, we have

$$\begin{aligned} R(A, z) &= \begin{pmatrix} R(A_1, z) & -R(A_1, z)CR(A_2, z) \\ 0 & R(A_2, z) \end{pmatrix} \\ &= \begin{pmatrix} R(A_1, z) & 0 \\ 0 & R(A_2, z) \end{pmatrix} + \begin{pmatrix} 0 & -R(A_1, z)CR(A_2, z) \\ 0 & 0 \end{pmatrix}. \end{aligned}$$

Therefore,

$$\|R(A, z)\| \le \max(\|R(A_1, z)\|, \|R(A_2, z)\|) + \|R(A_1, z)CR(A_2, z)\|.$$

Since $A_1X - XA_2 = C$, we have $(A_1 - zI)X - X(A_2 - zI) = C$. Consequently, $R(A_1, z)CR(A_2, z) = X\,R(A_2, z) - R(A_1, z)\,X$. This shows that

$$\|R(A_1, z)CR(A_2, z)\| = \|R(A_1, z)X - XR(A_2, z)\| \le 2r\,\max(\|R(A_1, z)\|, \|R(A_2, z)\|).$$

Hence we have $\|R(A, z)\| \le (2r+1)\max(\|R(A_1, z)\|, \|R(A_2, z)\|)$. Now, (a) follows by noting that $\|R(A, z)\| \ge \epsilon^{-1}$ implies

$$\max(\|R(A_1, z)\|,\, \|R(A_2, z)\|) \ge \frac{1}{(2r+1)\epsilon} = \frac{1}{g(\epsilon)}.$$

Next, we have $\|R(A,z)\| \le \max(\|R(A_1,z)\|, \|R(A_2,z)\|) + \|R(A_1,z)CR(A_2,z)\|$. Let $d := \max(\|R(A_1,z)\|, \|R(A_2,z)\|)$. Then $\|R(A,z)\| \le d + d^2c$. Therefore, if $\|R(A,z)\| \ge \epsilon^{-1}$, then $d^2c + d \ge \epsilon^{-1}$, that is, $\epsilon cd^2 + \epsilon d - 1 \ge 0$. This gives

$$d \ge \frac{-\epsilon + \sqrt{\epsilon^2 + 4\epsilon c}}{2\epsilon c} = \frac{2}{\epsilon + \sqrt{\epsilon^2 + 4\epsilon c}} = \frac{1}{f(\epsilon)}.$$

Hence (b) follows. ∎

For the 2-norm, the following result provides a better approximation of $\Lambda_\epsilon(A)$.

THEOREM 3.6 (Grammont and Largillier [14]). *Let* $\eta(\epsilon) := \epsilon\sqrt{1 + \frac{\|C\|_2}{\epsilon}}$. *Then for the* 2*-norm, we have*

$$\Lambda_\epsilon(A) \subset \Lambda_{\eta(\epsilon)}(A_1) \cup \Lambda_{\eta(\epsilon)}(A_2).$$

Note that if $\|X\|_2 \gg 1$, then $\|X\|_2 + \sqrt{\|X\|_2^2 + 1} \simeq 2\|X\|_2$. Therefore, in such a case, for the 2-norm, the approximations of $\Lambda_\epsilon(A)$ obtained, for example, by Theorems 3.4, 3.5(a), and 3.6 tend to be more or less the same.

4. Separations of matrices. Spectral overlap and separation of matrices play an important role in analyzing ϵ-stability of eigendecompositions [4]. We briefly describe these concepts, extend the notion of spectral overlap to the case of arbitrary norms, and establish relationships between separation of matrices and spectral overlap.

Let $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$. Then the separation of A and B, denoted by $\mathrm{sep}(A,B)$, is defined by

$$\mathrm{sep}(A,B) := \min\{\|AX - XB\| : \|X\| = 1\}. \tag{4.1}$$

Equivalently, if $\mathbf{T} : X \longmapsto AX - XB$ is the Sylvester operator, then (with respect to the induced operator norm of $\mathbf{T}$)

$$\mathrm{sep}(A,B) := \begin{cases} 1/\|\mathbf{T}^{-1}\|, & 0 \notin \sigma(\mathbf{T}), \\ 0, & 0 \in \sigma(\mathbf{T}). \end{cases}$$

For more on $\mathrm{sep}(A,B)$, we refer to [16], [15]. We denote sep with respect to the Frobenius norm by sep_F.

In 1979 Varah [19] introduced the notion of *spectral overlap* of matrices. He defined the spectral overlap of A and B, which we denote by $\mathrm{sep}_\lambda^{(1)}(A,B)$, by

$$\mathrm{sep}_\lambda^{(1)}(A,B) := \min_{\epsilon_1,\epsilon_2}\{\epsilon_1 + \epsilon_2 : \Lambda_{\epsilon_1}(A) \cap \Lambda_{\epsilon_2}(B) \ne \emptyset\},$$

where $\Lambda_\epsilon(A)$ is the 2-norm ϵ-pseudospectrum. This definition was motivated by the fact that if $\Lambda_\epsilon(A)$ and $\Lambda_\epsilon(B)$ overlap at λ, then there are $E \in \mathbb{C}^{m\times m}$ and $F \in \mathbb{C}^{n\times n}$ with $\|E\|_2 \le \epsilon$ and $\|F\|_2 \le \epsilon$ such that $\lambda \in \sigma(A+E) \cap \sigma(B+F)$. Subsequently, this definition was modified by Demmel in [4]. The modified spectral overlap, which we denote by $\mathrm{sep}_\lambda^{(2)}(A,B)$, is defined by

$$\mathrm{sep}_\lambda^{(2)}(A,B) := \inf\{\max(\sigma_{\min}(A - zI), \sigma_{\min}(B - zI)) : z \in \mathbb{C}\},$$

where $\sigma_{\min}(A)$ denotes the smallest singular value of A. It was stated in [4, p. 172] that $\mathrm{sep}_\lambda^{(2)}(A,B)$ is the magnitude of smallest perturbations of A and B required to induce a common eigenvalue. We prove a more general result in Proposition 4.2.

Since our aim is to analyze ϵ-stability of eigendecompositions for arbitrary operator norms, we further generalize the notion of spectral overlap to operator norms. We define

$$\operatorname{sep}(z, A) := \min\{\|E\| : z \in \sigma(A + E)\}. \tag{4.2}$$

Clearly $\Lambda_\epsilon(A) = \{z : \operatorname{sep}(z, A) \le \epsilon\}$. Here $\operatorname{sep}(z, A)$ plays the role of the smallest singular value of $A - zI$. For operator norms and the Frobenius norm, $\operatorname{sep}(z, A)$ and $\operatorname{sep}(B, A)$ given in (4.1) with $B = z$ are the same.

DEFINITION 4.1. *Let $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$. We define*

$$\operatorname{sep}_\lambda(A, B) := \inf\{\max(\operatorname{sep}(z, A),\ \operatorname{sep}(z, B)) : z \in \mathbb{C}\}.$$

Evidently, $\operatorname{sep}_\lambda(A, B) = \operatorname{sep}^{(2)}_\lambda(A, B)$ for the spectral and Frobenius norms. The following result shows that $\operatorname{sep}_\lambda(A, B)$ can be read off from the ϵ-pseudospectra of A and B.

PROPOSITION 4.2. *We have* $\operatorname{sep}_\lambda(A, B) = \min\{\epsilon : \Lambda_\epsilon(A) \cap \Lambda_\epsilon(B) \ne \emptyset\}$.

Proof. Let $\epsilon_0 := \min\{\epsilon : \Lambda_\epsilon(A) \cap \Lambda_\epsilon(B) \ne \emptyset\}$. Let z be a common point of $\Lambda_{\epsilon_0}(A)$ and $\Lambda_{\epsilon_0}(B)$. Then $\max\{\operatorname{sep}(z, A), \operatorname{sep}(z, B)\} \le \epsilon_0$. Taking infimum over z, we have $\operatorname{sep}_\lambda(A, B) \le \epsilon_0$. On the other hand, if $\epsilon < \epsilon_0$, then $\Lambda_\epsilon(A) \cap \Lambda_\epsilon(B) = \emptyset$. Therefore, for any $z \in \mathbb{C}$, we have $\max(\operatorname{sep}(z, A), \operatorname{sep}(z, B)) > \epsilon$. Taking infimum over z we have $\epsilon \le \operatorname{sep}_\lambda(A, B)$. Since $\epsilon < \epsilon_0$ is arbitrary, $\epsilon_0 \le \operatorname{sep}_\lambda(A, B)$. Hence the proof. ∎

Proposition 4.2 shows that $\operatorname{sep}_\lambda(A, B)$ is the smallest value of ϵ for which $\Lambda_\epsilon(A)$ and $\Lambda_\epsilon(B)$ have a common point. Therefore, for $\epsilon := \operatorname{sep}_\lambda(A, B)$, if $z \in \Lambda_\epsilon(A) \cap \Lambda_\epsilon(B)$, then $\operatorname{sep}(z, A) = \operatorname{sep}(z, B) = \operatorname{sep}_\lambda(A, B)$. Hence by (4.2) there are E and F such that $\|E\| = \|F\| = \operatorname{sep}_\lambda(A, B)$ and $z \in \sigma(A + E) \cap \sigma(B + F)$. This shows that $\operatorname{sep}_\lambda(A, B)$ is the magnitude of smallest perturbations of A and B required to induce a common eigenvalue.

Recall that $\Lambda_\epsilon(A) = \{z : \operatorname{sep}(z, A) \le \epsilon\}$. Therefore the boundary $\partial\Lambda_\epsilon(A)$ of $\Lambda_\epsilon(A)$ is a subset of $\Gamma := \{z : \operatorname{sep}(z, A) = \epsilon\}$. For operator norms, it is known that Γ is a closed curve or a union of closed curves (see [3], [7], [10]). Recently it has been shown in [13] that the same is true for any submultiplicative norm. Hence $\operatorname{sep}_\lambda(A, B)$ is the smallest value of ϵ for which $\Lambda_\epsilon(A)$ and $\Lambda_\epsilon(B)$ have a common boundary point. Thus a common eigenvalue induced by a smallest perturbation of A and B is a common boundary point of $\Lambda_\epsilon(A)$ and $\Lambda_\epsilon(B)$.

Since $\Lambda_\epsilon(A)$ is same for the 2-norm and the Frobenius norm, the proof of the following result is immediate.

COROLLARY 4.3. $\operatorname{sep}_\lambda$ *is the same for the* 2*-norm and the Frobenius norm.*

It is easy to see that $\operatorname{sep}_\lambda$ with respect to unitarily invariant norms is invariant under unitary similarity transformations of A and B. More generally, the following holds.

PROPOSITION 4.4. (a) *We have*

$$\frac{\operatorname{sep}_\lambda(A, B)}{\max(K(S), K(Q))} \le \operatorname{sep}_\lambda(SAS^{-1}, QBQ^{-1}) \le \max(K(S), K(Q))\operatorname{sep}_\lambda(A, B).$$

(b) *For a max-norm, we have*

$$\operatorname{sep}_\lambda(\operatorname{diag}(A_1, A_2, \ldots, A_k),\ \operatorname{diag}(B_1, B_2, \ldots, B_l)) = \min_{ij} \operatorname{sep}_\lambda(A_i, B_j).$$

Proof. For $z \in \mathbb{C}$, it is easy to see that $\operatorname{sep}(z, SAS^{-1}) \le K(S)\operatorname{sep}(z, A)$. Hence

$$\operatorname{sep}_\lambda(SAS^{-1}, QBQ^{-1}) \le \max(K(S), K(Q))\operatorname{sep}_\lambda(A, B).$$

Similarly, $\mathrm{sep}_\lambda(A,\, B) \le \max(K(S), K(Q))\mathrm{sep}_\lambda(SAS^{-1}, QBQ^{-1})$. Hence

$$\frac{\mathrm{sep}_\lambda(A,B)}{\max(K(S),K(Q))} \le \mathrm{sep}_\lambda(SAS^{-1}, QBQ^{-1}) \le \max(K(S),K(Q))\mathrm{sep}_\lambda(A,B).$$

The proof of (b) is immediate. ∎

The following result relates sep and sep_λ.

PROPOSITION 4.5. *We have* $\mathrm{sep}(A,B) \le 2\mathrm{sep}_\lambda(A,B)$. *If* $\dim(A) = 1$ *or* $\dim(B) = 1$, *then*

$$\mathrm{sep}_\lambda(A,B) \le \mathrm{sep}(A,B) \le 2\mathrm{sep}_\lambda(A,B).$$

Proof. There exist E and F with $\|E\| = \|F\| = \mathrm{sep}_\lambda(A,B)$ such that $A+E$ and $B+F$ have a common eigenvalue. Therefore, there exists X_0 with $\|X_0\| = 1$ such that $(A+E)X_0 - X_0(B+F) = 0$. Hence

$$\begin{aligned}\mathrm{sep}(A,B) &\le \|(A+E)X_0 - X_0(B+F) - (EX_0 - X_0F)\| \\ &\le \|EX_0 - X_0F\| \le \|E\| + \|F\| = 2\,\mathrm{sep}_\lambda(A,B).\end{aligned}$$

To prove the second part, we assume that $A = \mu$. Then from the definition,

$$\begin{aligned}\mathrm{sep}_\lambda(\mu,B) &= \inf\{\max(\mathrm{sep}(z,B),\, |z-\mu|) : z \in \mathbb{C}\} \\ &\le \max(\mathrm{sep}(\mu,B),\, 0) = \mathrm{sep}(\mu,B) = \mathrm{sep}(A,B).\end{aligned}$$

Hence the result follows. ∎

The results in Propositions 4.4 and 4.5 have been proved in [4] for $\mathrm{sep}_\lambda^{(2)}$. We have shown that the same results hold for general sep_λ as well.

If A and B are diagonal, then, for a max-norm, $\mathrm{sep}_\lambda(A,B) = \mathrm{dist}(\sigma(A),\, \sigma(B))/2$ and $\mathrm{sep}(A,B) = \mathrm{dist}(\sigma(A),\, \sigma(B))$, where $\mathrm{dist}(\sigma(A),\, \sigma(B)) := \min\{|\lambda - \mu| : \lambda \in \sigma(A),\, \mu \in \sigma(B)\}$. Hence if A and B are normal, then, for the 2-norm,

$$\mathrm{sep}_\lambda(A,B) = \mathrm{dist}(\sigma(A),\, \sigma(B))/2 = \mathrm{sep}(A,B)/2 = \mathrm{sep}_F(A,B)/2.$$

The following result shows that small perturbations in A and B induce a small change in $\mathrm{sep}_\lambda(A,B)$.

PROPOSITION 4.6. *We have*

$$\mathrm{sep}_\lambda(A,B) - \|E\| - \|F\| \le \mathrm{sep}_\lambda(A+E,\, B+F) \le \mathrm{sep}_\lambda(A,B) + \|E\| + \|F\|.$$

Proof. It is easy to see that $\mathrm{sep}(z,A) - \|E\| \le \mathrm{sep}(z, A+E) \le \mathrm{sep}(z,A) + \|E\|$ for all z and E. Hence the result follows. ∎

We conclude this section by providing a partial solution to a conjecture due to Demmel on the relationship between $\mathrm{sep}(A,B)$ and $\mathrm{sep}_\lambda(A,B)$. He conjectured (see [4, p. 183]) that if $\mathcal{A} := \left[\begin{smallmatrix} A & C \\ 0 & B \end{smallmatrix}\right]$ is such that $\|\mathcal{A}\|_F = 1$, then, for the Frobenius norm, there is a constant K such that

$$K \cdot (\mathrm{sep}_\lambda(A,B))^2 \le \mathrm{sep}_F(A,B).$$

If the boundary of $\Lambda_\epsilon(A)$ or $\Lambda_\epsilon(B)$ is rectifiable, that is, it has finite length, then we have the following result.

PROPOSITION 4.7. *Let* $\epsilon := \mathrm{sep}_\lambda(A,B) > 0$. *If the boundary of either* $\Lambda_\epsilon(A)$ *or* $\Lambda_\epsilon(B)$ *is rectifiable, then for an operator norm, we have*

$$\frac{2\pi}{L} \cdot (\mathrm{sep}_\lambda(A,B))^2 \le \mathrm{sep}(A,B),$$

where L *is the length of the boundary of* $\Lambda_\epsilon(A)$ *or* $\Lambda_\epsilon(B)$. *Set* $m := \min(\dim(A), \dim(B))$. *Then for the Frobenius norm, we have*

$$\frac{2\pi}{\sqrt{m}\,L} \cdot (\mathrm{sep}_\lambda(A,B))^2 \le \mathrm{sep}_F(A,B).$$

Proof. Assume that the boundary $\partial\Lambda_\epsilon(A)$ of $\Lambda_\epsilon(A)$ is rectifiable and let L be its length. Set $\Gamma := \partial\Lambda_\epsilon(A)$. First, we prove the result for an operator norm. Recall from (4.1) that $\mathrm{sep}(A,B) = 1/\|\mathbf{T}^{-1}\|$, where $\mathbf{T}(X) = AX - XB$. Since

$$\mathbf{T}^{-1}(X) = \frac{1}{2\pi i}\int_\Gamma R(A,z)XR(B,z)dz,$$

we have $\|\mathbf{T}^{-1}(X)\| \le L \max_{z\in\Gamma}(\|R(A,z)\|\ \|R(B,z)\|)\ \|X\|/2\pi$. Note that $\|R(A,z)\| = \epsilon^{-1}$ and $\|R(B,z)\| \le \epsilon^{-1}$ for $z \in \Gamma$. Therefore $\|\mathbf{T}^{-1}\| \le L/2\pi\epsilon^2$. Hence

$$\frac{2\pi}{L}(\mathrm{sep}_\lambda(A,B))^2 \le \mathrm{sep}(A,B).$$

To prove the result for the Frobenius norm, recall from (2.1) that the ϵ-pseudospectra for the 2-norm and the Frobenius norm are the same. Thus $\mathrm{sep}_\lambda(A,B)$ and L remain the same for the 2-norm and the Frobenius norm. Now the result follows by using the inequality $\mathrm{sep}_2(A,B) \le \sqrt{m}\,\mathrm{sep}_F(A,B)$, where sep_2 denotes sep with respect to the 2-norm. ∎

Obviously $\partial\Lambda_\epsilon(A)$ is rectifiable if A is normal. Although, we are not aware of a proof, for the 2-norm, the rectifiability of $\partial\Lambda_\epsilon(A)$ has been implicitly assumed by Trefethen [17] and Godunov [8]. Note, however, that for small ϵ, the components of $\Lambda_\epsilon(A)$ more or less look like deformed circles. As $\epsilon \to \mathrm{sep}_\lambda(A,B)$ some of these components may coalesce. After coalescence of components, $\Lambda_\epsilon(A)$ is expected to have a fairly regular shape and the boundary $\partial\Lambda_\epsilon(A)$ is expected to be rectifiable.

5. Lower bounds of geometric separation. Recall that a lower bound of gsep provides an upper bound of ϵ for ϵ-stability of eigendecompositions. The following theorem provides a general framework for obtaining lower bounds of gsep. Consider a matrix $A \in \mathbb{C}^{n\times n}$.

THEOREM 5.1. *Suppose that* A *is similar to* $\mathrm{diag}(A_1, A_2)$, *where* $\sigma(A_1)\cap\sigma(A_2) = \emptyset$. *Set* $\sigma_1 := \sigma(A_1)$. *If there exists a strictly increasing function* ϕ *such that*

$$\Lambda_\epsilon(A) \subset \Lambda_{\phi(\epsilon)}(A_1) \cup \Lambda_{\phi(\epsilon)}(A_2),$$

then $\phi^{-1}(\mathrm{sep}_\lambda(A_1,A_2)) \le \mathrm{gsep}(\sigma_1)$ *and* $\mathrm{sep}_\lambda(A_1,A_2) \le \phi(\mathrm{gsep}(\sigma_1))$.

Proof. Note that $\epsilon < \mathrm{gsep}(\sigma_1)$ if $\Lambda_{\phi(\epsilon)}(A_1)$ and $\Lambda_{\phi(\epsilon)}(A_2)$ are disjoint. But $\Lambda_{\phi(\epsilon)}(A_1)$ and $\Lambda_{\phi(\epsilon)}(A_2)$ remain disjoint if and only if $\phi(\epsilon) < \mathrm{sep}_\lambda(A_1,A_2)$. This shows that if $\phi(\epsilon) < \mathrm{sep}_\lambda(A_1,A_2)$, then $\epsilon < \mathrm{gsep}(\sigma_1)$. Since ϕ is strictly increasing, we have $\epsilon < \mathrm{gsep}(\sigma_1)$ whenever $\epsilon < \phi^{-1}(\mathrm{sep}_\lambda(A_1,A_2))$. Hence $\phi^{-1}(\mathrm{sep}_\lambda(A_1,A_2)) \le \mathrm{gsep}(\sigma_1)$ and $\mathrm{sep}_\lambda(A_1,A_2) \le \phi(\mathrm{gsep}(\sigma_1))$. ∎

Thus, in view of Theorem 5.1, various lower bounds of gsep are readily available from the inclusion theorems of $\Lambda_\epsilon(A)$. The Bauer–Fike–Wilkinson theorem (Theorem 3.2) immediately gives the following.

PROPOSITION 5.2. *Let A, κ, and s be as in Theorem* 3.2. *Let $\sigma_1 \subset \sigma(A)$. Then for an operator norm, we have*

$$\operatorname{gsep}(\sigma_1) \geq \frac{\operatorname{dist}(\sigma_1,\, \sigma(A) \setminus \sigma_1)}{2s}.$$

For a max-norm, we have $\operatorname{gsep}(\sigma_1) \geq \frac{\operatorname{dist}(\sigma_1, \sigma(A)\setminus\sigma_1)}{2\kappa}$.

For general matrices, we have the following bounds.

THEOREM 5.3. *Let A, κ, and s be as in Theorem* 3.3. *Set $\sigma_j := \sigma(A_j)$. Then for a max-norm, we have*

$$\operatorname{gsep}(\sigma_1, \ldots, \sigma_m) \geq \frac{\operatorname{sep}_\lambda(A_1, \ldots, A_m)}{\kappa}.$$

In particular, for the 2*-norm, we have*

$$\operatorname{gsep}(\sigma_1, \ldots, \sigma_m) \geq \frac{\operatorname{sep}_\lambda(A_1, \ldots, A_m)}{s}.$$

Proof. By Theorem 3.3, we have $\phi(\epsilon) = \kappa\epsilon$ for the first case and $\phi(\epsilon) = s\epsilon$ for the second case. Hence the result follows from Theorem 5.1. ∎

Now, for the rest of this section, we assume that A is given by (1.3). Recall that P is the spectral projection associated with A and $\sigma_1 := \sigma(A_1)$, and that $A_1X - XA_2 = C$. By Theorems 3.4 and 5.1, we have the following bound.

THEOREM 5.4 (Demmel [4]). *For the* 2*-norm, we have*

$$\operatorname{gsep}(\sigma_1) \geq \frac{\operatorname{sep}_\lambda(A_1, A_2)}{\|P\|_2 + \sqrt{\|P\|_2^2 - 1}} = \frac{\operatorname{sep}_\lambda(A_1, A_2)}{\|X\|_2 + \sqrt{\|X\|_2^2 + 1}}. \tag{5.1}$$

A slightly weaker bound given by

$$\operatorname{gsep}(\sigma_1) \geq \frac{\operatorname{sep}_\lambda(A_1, A_2)}{2\|P\|_2} \tag{5.2}$$

is due to Wilkinson [22] and follows from Theorem 5.3. When $\|P\|_2 \simeq 1$, (5.1) provides a much better estimate of gsep than (5.2). A more conservative bound given by

$$\operatorname{gsep}(\sigma_1) \geq \frac{\operatorname{sep}_F(A_1, A_2)}{4\|P\|_2} \tag{5.3}$$

is due to Stewart [16] and follows from (5.2). Indeed, since $\operatorname{sep}_F(A_1, A_2) \leq 2\operatorname{sep}_\lambda(A_1, A_2)$, from (5.2) we have

$$\operatorname{gsep}(\sigma_1) \geq \frac{\operatorname{sep}_\lambda(A_1, A_2)}{2\|P\|_2} \geq \frac{\operatorname{sep}_F(A_1, A_2)}{4\|P\|_2}.$$

For a max-norm, the following bounds hold.

THEOREM 5.5. *Let $A_1X - XA_2 = C$. Then for a max-norm, we have*

(a) $\operatorname{gsep}(\sigma_1) \geq \frac{\operatorname{sep}_\lambda(A_1,A_2)}{2\|X\|+1}$,

(b) $\operatorname{gsep}(\sigma_1) \geq \frac{(\operatorname{sep}_\lambda(A_1,A_2))^2}{\|C\|+\operatorname{sep}_\lambda(A_1,A_2)}$.

Proof. From Theorem 3.5(a), we have $\phi(\epsilon) = g(\epsilon) = (2\|X\| + 1)\epsilon$. Therefore, by Theorem 5.1, we have $\mathrm{gsep}(\sigma_1) \geq \mathrm{sep}_\lambda(A_1, A_2)/(2\|X\| + 1)$. Next, Theorem 3.5(b) gives $\phi(\epsilon) = f(\epsilon) = (\epsilon + \sqrt{\epsilon^2 + 4\,\|C\|\,\epsilon})/2$. Since ϕ is strictly increasing on $[0,\ \infty)$ and $\phi^{-1}(\epsilon) = \epsilon^2/(\epsilon + \|C\|)$, the result follows from Theorem 5.1. ∎

For the 2-norm, the following bounds provide better estimates of gsep.

THEOREM 5.6. *Set* $\mathrm{sep}_\lambda := \mathrm{sep}_\lambda(A_1, A_2)$. *Let* sep *and* sep_F *denote* $\mathrm{sep}(A_1, A_2)$ *for the* 2*-norm and the Frobenius norm, respectively. Then for the* 2*-norm and the Frobenius norm, the following bounds hold:*

$$\text{(5.4)} \quad \text{(a)} \qquad \mathrm{gsep}(\sigma_1) \geq \frac{2\,\mathrm{sep}_\lambda^2}{\|C\|_2 + \sqrt{\|C\|_2^2 + 4\,\mathrm{sep}_\lambda^2}} \geq \frac{\mathrm{sep}^2}{2(\|C\|_2 + \sqrt{\|C\|_2^2 + \mathrm{sep}^2})},$$

$$\text{(5.5)} \quad \text{(b)} \qquad \mathrm{gsep}(\sigma_1) \geq \frac{2\,\mathrm{sep}_\lambda^2}{\|C\|_F + \sqrt{\|C\|_F^2 + 4\,\mathrm{sep}_\lambda^2}} \geq \frac{\mathrm{sep}_F^2}{2(\|C\|_F + \sqrt{\|C\|_F^2 + \mathrm{sep}_F^2})}.$$

Proof. By Theorem 3.6, we have $\phi(\epsilon) = \epsilon\sqrt{1 + \|C\|_2/\epsilon}$. It is easy to see that ϕ is strictly increasing on $[0, \infty)$ and $\phi^{-1}(\epsilon) = 2\epsilon^2/(\|C\|_2 + \sqrt{\|C\|_2^2 + 4\epsilon^2})$. Hence the first inequality in (a) follows from Theorem 5.1. The first inequality in (b) follows from the fact that $\|C\|_2 \leq \|C\|_F$. The second inequalities in (a) and (b) follow from the fact that $\mathrm{sep}(A_1, A_2) \leq 2\,\mathrm{sep}_\lambda(A_1, A_2)$. ∎

It is well known (see [4], [19]) that $\|P\|_2 \leq \left(1 + \|C\|_F^2/\mathrm{sep}_F^2\right)^{1/2}$. Hence the second inequality in (5.5) also follows from (5.1). More generally, as a consequence of Theorems 3.1 and 5.1, we have the following bounds.

PROPOSITION 5.7. *Suppose that* $A \in \mathbb{C}^{n\times n}$ *is similar to* $\mathrm{diag}(A_1, A_2)$, *where* $\sigma(A_1) \cap \sigma(A_2) = \emptyset$. *Set* $\sigma_1 := \sigma(A_1)$. *Then for a max-norm as well as for the Frobenius norm, we have*

$$\mathrm{gsep}(\sigma_1) \geq \frac{\mathrm{sep}_\lambda(A_1, A_2)}{\inf_S K(S)} \geq \frac{\mathrm{sep}(A_1, A_2)}{2\inf_S K(S)},$$

where S *is such that* $S^{-1}AS = \mathrm{diag}(A_1, A_2)$.

Let (5.4a) refer to the lower bound in the first inequality in (5.4). The lower bound (5.1) is known to provide the best estimate of gsep. We now show that (5.4a) compares well with (5.1). Consider the map $\mathbf{T} : X \mapsto A_1X - XA_2$. If C is such that $\|X\|_2 = \|\mathbf{T}^{-1}C\|_2 = \|\mathbf{T}^{-1}\|\,\|C\|_2 = \|C\|_2/\mathrm{sep}$, then $\|X\|_2 = \|C\|_2/\mathrm{sep} \geq \|C\|_2/2\mathrm{sep}_\lambda$. Hence it follows that (5.4a) is a better lower bound of gsep than (5.1). It is natural to ask, *How much better could* (5.4a) *be over* (5.1) *and vice versa?*[1] The following result shows that they can differ from each other at most by a factor of sep_λ.

THEOREM 5.8. *Let* τ *be a constant such that* $\tau\,\mathrm{sep}_\lambda^2 \leq \mathrm{sep}$. *Then either*

$$(5.1) \geq (5.4a) \geq \frac{2\mathrm{sep}_\lambda}{\|\mathbf{T}\|}(5.1) \quad \textit{or} \quad (5.4a) \geq (5.1) \geq \frac{\tau\,\mathrm{sep}_\lambda}{2}(5.4a).$$

Proof. First, note that $\tau\mathrm{sep}_\lambda^2 \leq \mathrm{sep} \leq 2\mathrm{sep}_\lambda$. Hence $\tau\mathrm{sep}_\lambda \leq 2$. Set $c := \|C\|_2$. Then $c/\|\mathbf{T}\| \leq \|X\|_2 \leq c/\mathrm{sep}$. Since sep is bounded below by $\tau\mathrm{sep}_\lambda^2$ and bounded above by $2\mathrm{sep}_\lambda$, we have two cases: either $c/\|\mathbf{T}\| \leq \|X\|_2 < c/2\mathrm{sep}_\lambda$ or $c/2\mathrm{sep}_\lambda \leq \|X\|_2 \leq c/\tau\mathrm{sep}_\lambda^2$. As noted above, if $\|X\|_2 = c/\mathrm{sep}$, then $\|X\|_2$ satisfies the second case.

[1] The authors thank one of the referees for raising this question.

Suppose that $c/\|\mathbf{T}\| \le \|X\|_2 < c/2\text{sep}_\lambda$. Let $\kappa := 2\text{sep}_\lambda/\|\mathbf{T}\|$. Then $\kappa \cdot c/2\text{sep}_\lambda \le \|X\|_2 < c/2\,\text{sep}_\lambda$. Hence $\kappa < 1$ and

$$\kappa \cdot \left(c/2\text{sep}_\lambda + \sqrt{c^2/4\,\text{sep}_\lambda^2 + 1} \right) \le \|X\|_2 + \sqrt{\|X\|_2^2 + 1} \le c/2\,\text{sep}_\lambda + \sqrt{c^2/4\,\text{sep}_\lambda^2 + 1}.$$

Consequently, we have $\frac{1}{\kappa}(5.4\text{a}) \ge (5.1) \ge (5.4\text{a})$, which in turn gives

$$(5.1) \ge (5.4\text{a}) \ge \frac{2\text{sep}_\lambda}{\|\mathbf{T}\|}(5.1).$$

Next, suppose that $\|X\|_2$ satisfies the second case. Since $2/\tau\text{sep}_\lambda \ge 1$, we have

$$\begin{aligned} c/2\text{sep}_\lambda + \sqrt{c^2/4\text{sep}_\lambda^2 + 1} &\le \|X\|_2 + \sqrt{\|X\|_2^2 + 1} \\ &\le 2/\tau\text{sep}_\lambda \cdot \left(c/2\text{sep}_\lambda + \sqrt{c^2/4\,\text{sep}_\lambda^2 + 1} \right). \end{aligned}$$

Hence $(5.4\text{a}) \ge (5.1) \ge \frac{\tau\text{sep}_\lambda}{2}(5.4\text{a})$. ∎

Since $\|X\|_2 \le c/\text{sep} \le c/\tau\text{sep}_\lambda^2$, it follows that $(5.1) \ge \frac{\tau\text{sep}_\lambda}{2}(5.4\text{a})$ always holds. We now illustrate by numerical examples that (5.1) could be smaller than (5.4a) by a factor of sep_λ, and vice versa. We assume that A is given by (1.3).

Example 5.1. Let

$$A_1 := \begin{bmatrix} 0.5 & 50 & 30 \\ 0 & 0.5 & 50 \\ 0 & 0 & 0.5 \end{bmatrix}, \qquad A_2 := \begin{bmatrix} -0.5 & -20 & -80 \\ 0 & -0.5 & -20 \\ 0 & 0 & -0.5 \end{bmatrix}$$

and let C be the following matrix:

$$\begin{bmatrix} -5.962981568242861 \times 10^{-5} & 3.725569560589521 \times 10^{-6} & -1.860207631770143 \times 10^{-7} \\ 9.683970645185497 \times 10^{-3} & -3.723353543788620 \times 10^{-4} & 9.302840353024797 \times 10^{-6} \\ -9.995693003184436 \times 10^{-1} & 2.769622874989600 \times 10^{-2} & -4.595614058539184 \times 10^{-4} \end{bmatrix}.$$

For various values of ϵ, Figure 5.1 shows portions of the contour plots of $\Lambda_\epsilon(\text{diag}(A_1, A_2))$ and $\Lambda_\epsilon(A)$ where the components coalesce. From the left plot we have $\text{sep}_\lambda = 1.122 \times 10^{-4}$ and from the right plot $\text{gsep} = 1.7466 \times 10^{-8}$. For these matrices, we have $\text{sep}_F(A_1, A_2) = 1.861 \times 10^{-7}$ showing that $\text{sep}_F = \mathcal{O}(\text{sep}_\lambda^2)$. The first row of Table 5.1 gives the values of gsep, (5.4a), and (5.1). These results demonstrate that (5.1) is smaller than (5.4a) by a factor of sep_λ. Notice that (5.4a) provides a sharp estimate of gsep, whereas the estimate provided by (5.1) is quite conservative. ∎

Next, we illustrate that (5.4a) could be smaller than (5.1) by a factor of sep_λ.

Example 5.2. Let A_1 and A_2 be as in Example 5.1 and let C be the matrix

$$\begin{bmatrix} -3.976344050589186 \times 10^{-1} & -1.919161875563033 \times 10^{-1} & -4.203140565201795 \times 10^{-1} \\ -1.866525471931694 \times 10^{-1} & -1.988322234548340 \times 10^{-1} & -6.650420651861800 \times 10^{-1} \\ -3.618542151771120 \times 10^{-3} & -7.402307617753745 \times 10^{-2} & -3.259513248630305 \times 10^{-1} \end{bmatrix}.$$

The values of $\text{sep}_F(A_1, A_2)$ and $\text{sep}_\lambda(A_1, A_2)$ are the same as in Example 5.1. The second row of Table 5.1 gives the values of gsep, (5.4a), and (5.1). Notice that (5.4a) is smaller than (5.1) by a factor of sep_λ. Obviously, (5.4a) provides a conservative estimate of gsep, whereas the estimate provided by (5.1) is sharp. ∎

Finally, we illustrate the limitations of these bounds. The following example shows that both (5.4a) and (5.1) provide conservative estimates of gsep.

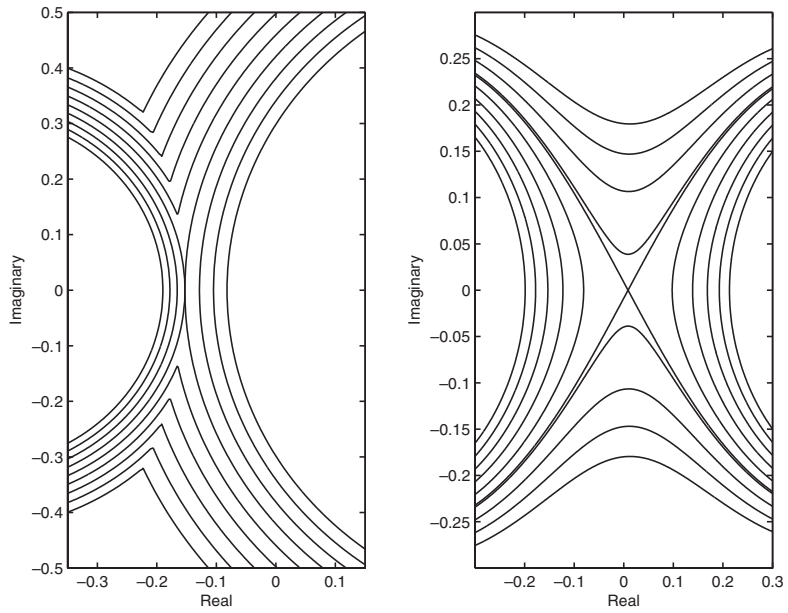


FIG. 5.1. *The left figure shows a portion of the contour plot of* $\Lambda_\epsilon(\mathrm{diag}(A_1, A_2))$*, and the right figure shows a portion of the contour plot of* $\Lambda_\epsilon(A)$ *for various values of* ϵ*. In the left figure the components coalesce for* $\epsilon = 1.122 \times 10^{-4}$*, and in the right figure the components coalesce for* $\epsilon = 1.7466 \times 10^{-8}$*, thus giving* sep_λ *and* gsep*, respectively.*

TABLE 5.1

Values of $\mathrm{gsep}(\sigma(A_1))$ *with respect to the* 2*-norm,* (5.4a)*, and* (5.1) *for Examples* 5.1, 5.2, 5.3, *and* 5.4*. The first and second rows illustrate that these lower bounds could be smaller than each other by a factor of* sep_λ*. The third row illustrates that both* (5.4a) *and* (5.1) *provide conservative estimates of* gsep*. The last three rows show that* (5.4a) *and* (5.1) *provide sharp estimates of* gsep.

Example	gsep	(5.4a)	(5.1)
5.1	1.7466×10^{-8}	1.2589×10^{-8}	1.044×10^{-11}
5.2	1.1171×10^{-4}	1.3016×10^{-8}	1.1127×10^{-4}
5.3	2.557×10^{-11}	1.2552×10^{-13}	6.8084×10^{-15}
5.4(a)	1.5456×10^{-2}	1.4253×10^{-2}	1.1867×10^{-2}
5.4(b)	1.48×10^{-2}	1.2×10^{-2}	1.28×10^{-2}
5.4(c)	3.15×10^{-2}	1.96×10^{-2}	1.47×10^{-2}

Example 5.3. Let

$$A_1 := \begin{bmatrix} 1 & 1 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 + 10^{-8} \end{bmatrix}, \qquad A_2 := \begin{bmatrix} 1 & 10^{-4} \\ 2 & 1 \end{bmatrix},$$

and

$$C := \begin{bmatrix} 10 & 7 \\ 8 & 3 \\ 0 & 1 \end{bmatrix}.$$

We have $\mathrm{sep}_F(A_1, A_2) = 10^{-8}$ and $\mathrm{sep}_\lambda(A_1, A_2) = 1.3632 \times 10^{-6}$. The third row of Table 5.1 gives the values of gsep, (5.4a) and (5.1). Observe that although (5.4a) is certainly better than (5.1), both bounds provide conservative estimates of gsep. ∎

TABLE 5.2
The first three rows give values of gsep *and the lower bounds in Theorem* 5.5 *for the* 1*-norm. The last three rows give values of* gsep *and the lower bounds in Theorem* 5.5 *for the* ∞*-norm.*

Example	gsep_1	Theorem 5.5(a)	Theorem 5.5(b)
5.4(a)	1.1932×10^{-1}	7.41×10^{-3}	1.019×10^{-2}
5.4(b)	7.55×10^{-2}	6.1×10^{-3}	6.5×10^{-3}
5.4(c)	2.63×10^{-2}	8.7×10^{-3}	1.29×10^{-2}
Example	gsep_∞	Theorem 5.5(a)	Theorem 5.5(b)
5.4(a)	1.371×10^{-2}	1.018×10^{-2}	1.051×10^{-2}
5.4(b)	1.128×10^{-1}	1.33×10^{-2}	6.5×10^{-3}
5.4(c)	5.84×10^{-2}	1.71×10^{-2}	3.5×10^{-2}

The first three rows of Table 5.1 demonstrate the limitations of the lower bounds of gsep. We, however, mention that these examples have been specially constructed to expose the weaknesses of these bounds. Our contention is that, generically, these lower bounds provide good approximations of gsep. We illustrate this by considering an example. The matrices in the following example have been chosen almost arbitrarily.

Example 5.4. (a) Consider $A_1 := \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$, $A_2 := 0.5$, and $C := \left[\begin{smallmatrix} 0.4 \\ -1 \end{smallmatrix}\right]$.

(b) Next, consider

$$A_1 := \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad A_2 := \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}, \quad C := \begin{bmatrix} 0 & 1 \\ 2 & 1 \\ 4 & 2 \end{bmatrix}.$$

(c) Finally, consider

$$A_1 := \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 2 \\ 2 & 0 & 3 \end{bmatrix}, \quad A_2 := \begin{bmatrix} 3 & 1 \\ 0 & 1 \end{bmatrix}, \quad C := \begin{bmatrix} 1 & 3 \\ 0 & 6 \\ 7 & 0 \end{bmatrix}.$$

For these matrices, the last three rows of Table 5.1 show that the lower bounds (5.4a) and (5.1) indeed provide sharp estimates of gsep. ∎

For $p = 1$ and $p = \infty$, let gsep_p denote $\text{gsep}(\sigma(A_1))$ with respect to the p-norm. Table 5.2 gives relevant quantities for the 1-norm and the ∞-norm.

We conclude this section with a final note. Generically, the lower bounds discussed above may provide good approximations of gsep. As we shall see, for 2-by-2 matrices, gsep attains some of these bounds. The lower bounds in the second inequalities of (5.4) and (5.5), respectively, are expected to be smaller than those in the first inequalities by an order of magnitude whenever $\text{sep} = \mathcal{O}(\text{sep}_\lambda^2)$. The same is true for the lower bound (5.3). As we have already seen, these bounds have their limitations and hence too much should not be read into them. Whether it is our bounds or Demmel's, or for that matter any bounds obtained from Theorem 5.1, in general, all these lower bounds are expected to provide poor approximations of gsep unless $\phi(\epsilon) \simeq \beta\epsilon$ for some modest size $\beta > 1$, that is, unless the eigendecomposition is well-conditioned. Nevertheless, most often these bounds provide a good measure of the extent of ill-conditioning of the eigendecompositions and hence may serve the desired purpose. If an application requires a reliable and accurate information about ill-conditioning of an eigendecomposition, then in such a case the computation of gsep is the best option.

6. Relationship between gsep, sep, sep_λ, and diss. The notion of *dissociation* of eigenvalues was introduced in [4] in order to analyze ϵ-stability of eigendecompositions. The dissociation of an eigenvalue $\lambda \in \sigma(A)$ from the rest of $\sigma(A)$,

denoted by $\mathrm{diss}(\lambda)$, is the smallest value of $\|E\|$ for which λ coalesces with an eigenvalue $\mu \in \sigma(A) \setminus \{\lambda\}$ as $A \to A + E$. Similarly, the dissociation of $\sigma_1 \subset \sigma(A)$ from the rest of $\sigma(A)$, denoted by $\mathrm{diss}(\sigma_1)$, is the smallest value of $\|E\|$ for which an eigenvalue $\lambda \in \sigma_1$ coalesces with an eigenvalue $\mu \in \sigma(A) \setminus \sigma_1$ as $A \to A + E$. For more on diss we refer to [4]. Clearly an eigendecomposition $\sigma(A) = \cup_{j=1}^{m} \sigma_j$ is ϵ-stable if and only if $\epsilon < \min_{1 \le j \le m} \mathrm{diss}(\sigma_j)$ [4]. The main drawback of this approach is that except for some special cases it is not known how to compute diss. Nevertheless a lower bound of diss provides a sufficient condition for ϵ-stability of eigendecompositions, and the main effort in [4] has been towards obtaining the lower bounds.

We denote diss for the Frobenius norm and the 2-norm by diss_F and diss_2, respectively. Also, when it is necessary to show the dependence of diss on A, we write $\mathrm{diss}(\sigma_1, A)$ instead of $\mathrm{diss}(\sigma_1)$.

We have already seen that the pseudospectra-based approach to analyzing ϵ-stability of eigendecompositions leads to the notion of geometric separation of eigenvalues. Now, the question whether gsep characterizes ϵ-stability of eigendecompositions is equivalent to asking whether $\mathrm{gsep}(\sigma_1) = \mathrm{diss}(\sigma_1)$. This issue has been analyzed extensively in [1], [3], where it has been shown that the equality holds for the 2-norm and, under appropriate assumption, for the Frobenius norm. However, for other norms, the problem is still unsolved. In what follows, we show that gsep provides a best possible lower bound of diss, and to see how much gsep may differ from diss, we obtain upper bound of diss in terms of gsep.

Notice that for an eigenvalue $\lambda \in \sigma(A)$ to coalesce with another eigenvalue $\mu \in \sigma(A)$, it is essential for the components of $\Lambda_\epsilon(A)$ containing λ and μ to coalesce. This immediately gives

$$\mathrm{gsep}(\sigma_1) \le \mathrm{diss}(\sigma_1).$$

Since $\Lambda_\epsilon(A)$ provides the best possible localization for eigenvalues of A when A varies in $\mathbf{A}(\epsilon)$, the lower bound provided by gsep is clearly better than any bounds obtained by approximate localization of eigenvalues of A.

Now, assume that A is given by (1.3). Set $\sigma_1 := \sigma(A_1)$.

PROPOSITION 6.1. *For a max-norm and for the Frobenius norm, respectively, we have*

$$\mathrm{diss}(\sigma_1) \le \mathrm{sep}_\lambda(A_1, A_2) \quad \textit{and} \quad \mathrm{diss}_F(\sigma_1) \le \sqrt{2}\,\mathrm{sep}_\lambda(A_1, A_2).$$

Proof. From the definition of $\mathrm{sep}_\lambda(A_1, A_2)$, there are E_1 and E_2 with $\|E_1\| = \|E_2\| = \mathrm{sep}_\lambda(A_1, A_2)$ such that $A_1 + E_1$ and $A_2 + E_2$ have a common eigenvalue. Taking $E := \mathrm{diag}(E_1, E_2)$, the desired results follow. ∎

Thus, in view of Proposition 6.1 and Theorem 5.1, we immediately obtain the following general relationship between $\mathrm{gsep}, \mathrm{sep}_\lambda, \mathrm{diss}$, and sep.

THEOREM 6.2. *Let $\phi(\epsilon)$ be a strictly increasing function such that*

$$\Lambda_\epsilon(A) \subset \Lambda_{\phi(\epsilon)}(A_1) \cup \Lambda_{\phi(\epsilon)}(A_2).$$

(a) *Then we have* $\mathrm{sep}(A_1, A_2) \le 2\,\mathrm{sep}_\lambda(A_1, A_2) \le 2\,\phi(\mathrm{gsep}(\sigma_1))$.

(b) *For a max-norm, we have*

$$\phi^{-1}(\mathrm{sep}_\lambda(A_1, A_2)) \le \mathrm{gsep}(\sigma_1) \le \mathrm{diss}(\sigma_1) \le \mathrm{sep}_\lambda(A_1, A_2) \le \phi(\mathrm{gsep}(\sigma_1)).$$

In particular, if $A_1 = \mu$, then

$$\phi^{-1}(\mathrm{sep}_\lambda(\mu, A_2)) \le \mathrm{gsep}(\mu) \le \mathrm{diss}(\mu) \le \mathrm{sep}_\lambda(\mu, A_2) \le \mathrm{sep}(\mu, A_2) \le 2\,\phi(\mathrm{gsep}(\mu)).$$

As a consequence of Theorems 6.2 and 3.1, we have the following.

PROPOSITION 6.3. *For a max-norm, we have*

$$\frac{\operatorname{sep}_\lambda(A_1, A_2)}{\inf_S K(S)} \le \operatorname{gsep}(\sigma_1) \le \operatorname{diss}(\sigma_1) \le \operatorname{sep}_\lambda(A_1, A_2) \le \inf_S K(S) \operatorname{gsep}(\sigma_1),$$

where $S^{-1}AS = \operatorname{diag}(A_1, A_2)$. *Also* $\operatorname{sep}(A_1, A_2) \le 2\operatorname{sep}_\lambda(A_1, A_2) \le 2 \inf_S K(S) \operatorname{gsep}(\sigma_1)$.

In particular, if $A_0 := \operatorname{diag}(A_1, A_2)$ *with* $\sigma(A_1) \cap \sigma(A_2) = \emptyset$ *and* $\sigma_1 := \sigma(A_1)$, *then*

$$\operatorname{gsep}(\sigma_1, A_0) = \operatorname{diss}(\sigma_1, A_0) = \operatorname{sep}_\lambda(A_1, A_2).$$

The above result shows that all three concepts, namely, gsep, diss, and $\operatorname{sep}_\lambda$, coincide for block diagonal matrices. This means that for analyzing the coalescence of eigenvalues of block diagonal matrices, it is enough to restrict ourselves to block diagonal perturbations. The following bounds follow from Theorems 3.5, 3.6, and 6.2.

PROPOSITION 6.4. *Let* $A_1X - XA_2 = C$. *Then for a max-norm,*

$$\operatorname{gsep}(\sigma_1) \le \operatorname{diss}(\sigma_1) \le (2\|X\| + 1) \operatorname{gsep}(\sigma_1),$$
$$\operatorname{gsep}(\sigma_1) \le \operatorname{diss}(\sigma_1) \le \frac{\operatorname{gsep}(\sigma_1) + \sqrt{(\operatorname{gsep}(\sigma_1))^2 + 4\|C\| \operatorname{gsep}(\sigma_1)}}{2}.$$

For the 2*-norm and the Frobenius norm, we have*

$$\operatorname{gsep}(\sigma_1) \le \operatorname{diss}_2(\sigma_1) \le \operatorname{gsep}(\sigma_1)\sqrt{1 + \frac{\|C\|_2}{\operatorname{gsep}(\sigma_1)}},$$
$$\operatorname{gsep}(\sigma_1) \le \operatorname{diss}_F(\sigma_1) \le \sqrt{2}\operatorname{gsep}(\sigma_1)\sqrt{1 + \frac{\|C\|_2}{\operatorname{gsep}(\sigma_1)}}.$$

Similarly, for the 2-norm, the bound

$$\operatorname{gsep}(\sigma_1) \le \operatorname{diss}_2(\sigma_1) \le (\|P\|_2 + \sqrt{\|P\|_2^2 - 1}) \cdot \operatorname{gsep}(\sigma_1)$$

follows from Theorems 6.2 and 3.4. Thus if an estimate of gsep is available, then the above bounds provide intervals which localize diss. Most often, the above localization of diss by gsep may be quite effective.

7. Bounds on gsep and diss for 2-by-2 matrices. Let $A := \left[\begin{smallmatrix} a & d \\ 0 & b \end{smallmatrix}\right]$, where $a \ne b$. For the matrix A, it is shown in [4, Lemma 5.7] that the lower bound (5.1) is equal to diss_2. A simple proof of this fact follows from a result of Wilkinson.

PROPOSITION 7.1 (Wilkinson [21]). *There is a rank one matrix* E *such that*

$$\|E\|_2 = \|E\|_F = \frac{|a-b|^2}{2(|d| + \sqrt{|d|^2 + |a-b|^2})}$$

and $\sigma(A+E) = \{\frac{a+b}{2}\}$. *Hence* $\operatorname{diss}_2 \le \operatorname{diss}_F \le \frac{|a-b|^2}{2(|d|+\sqrt{|d|^2+|a-b|^2})}$.

Since $A_1 := a$ and $A_2 := b$, by Theorem 5.6, we have

$$\frac{\operatorname{sep}_\lambda}{\|P\|_2 + \sqrt{\|P\|_2^2 - 1}} = \frac{2\operatorname{sep}_\lambda^2}{\|C\|_2 + \sqrt{\|C\|_2^2 + 4\operatorname{sep}_\lambda^2}} = \frac{|a-b|^2}{2(|d| + \sqrt{|d|^2 + |a-b|^2})},$$

and hence $\frac{|a-b|^2}{2(|d|+\sqrt{|d|^2+|a-b|^2})} \le \operatorname{gsep} \le \operatorname{diss}_2 \le \operatorname{diss}_F$. This proves the following.

THEOREM 7.2. *We have* $\mathrm{diss}_2 = \mathrm{diss}_F = \mathrm{gsep} = \frac{|a-b|^2}{2(|d|+\sqrt{|d|^2+|a-b|^2})}$.

Thus for 2-by-2 matrices, the lower bounds in Theorem 5.6 are attained by gsep and diss. By contrast, the bounds in Theorem 5.5 may not be attained by gsep. We show this by an example. First, note that for the 1-norm and the ∞-norm,

$$\frac{\mathrm{sep}_\lambda}{2\,\|X\|+1} = \frac{\mathrm{sep}_\lambda^2}{\|C\|+\mathrm{sep}_\lambda} = \frac{|a-b|^2}{4|d|+2|a-b|}.$$

To see that this value is not attained by gsep, consider $A := \left[\begin{smallmatrix} 11 & 4 \\ 0 & 1 \end{smallmatrix}\right]$. Then for the 1-norm and the ∞-norm, $\mathrm{gsep} = 3.03336$, whereas $\frac{|a-b|^2}{4|c|+2|a-b|} = 2.77778$.

It is easy to see [21] that the perturbation $E := -\frac{(a-b)^2}{4\,c} e_2 e_1^T$ induces a double eigenvalue of $A+E$ at $(a+b)/2$, where $e_1 := [1,0]^t$ and $e_2 := [0,1]^t$. Hence for the 1-norm and the ∞-norm, we have

$$\mathrm{gsep} \le \mathrm{diss} \le \frac{|a-b|^2}{4\,|d|}.$$

In order to obtain a sharp localization of diss, we construct a matrix E such that $\|E\|_1 = \|E\|_\infty = \frac{3}{2}\left(\frac{|a-b|^2}{2|a-b|+4|d|}\right)$ and $\sigma(A+E) = \{\frac{a+b}{2}\}$.

PROPOSITION 7.3. *Let* $E := -\frac{|a-b|^2}{2|a-b|+4|d|}\left[\begin{smallmatrix} e^{i\theta_1} \\ 2e^{i\theta_2} \end{smallmatrix}\right]\left[\begin{smallmatrix} \frac{1}{2}, & -\frac{1}{4}e^{i(\theta_1-\theta_2)} \end{smallmatrix}\right]$, *where* $\theta_1 = 2\pi - \arg(\frac{1}{a-b})$, *and* $\theta_2 = 2\pi - \arg(\frac{d}{(a-b)^2})$. *Then* $\sigma(A+E) = \{\frac{a+b}{2}\}$.

Proof. Note that $\|E\|_1 = \|E\|_\infty = \frac{3}{2}\frac{|a-b|^2}{2|a-b|+4|d|}$ and $\mathrm{trace}(E) = 0$. Therefore, $\mathrm{trace}(A+E) = \mathrm{trace}(A) = a+b$. Hence to show that $\sigma(A+E) = \{\frac{a+b}{2}\}$, it is enough to prove that $\frac{a+b}{2} \in \sigma(A+E)$. Set $\lambda := (a+b)/2$. Then

$$R(A,\lambda) = \begin{bmatrix} \dfrac{2}{a-b} & \dfrac{4d}{(a-b)^2} \\ 0 & -\dfrac{2}{a-b} \end{bmatrix}.$$

This shows that $\|R(A,\lambda)\|_1 = \|R(A,\lambda)\|_\infty = \frac{2}{|a-b|} + \frac{4|d|}{|a-b|^2}$ and

$$\left[\frac{1}{2}, -\frac{1}{4}e^{i(\theta_1-\theta_2)}\right] R(A,\lambda) \left[e^{i\theta_1} 2e^{i\theta_2}\right] = \frac{2}{|a-b|} + \frac{4|d|}{|a-b|^2}.$$

Since

$$\frac{|a-b|^2}{2|a-b|+4|d|} = 1/\|R(A,\lambda)\|_1 = 1/\|R(A,\lambda)\|_\infty,$$

setting $u := R(A,\lambda)\left[\begin{smallmatrix} e^{i\theta_1} \\ 2e^{i\theta_2} \end{smallmatrix}\right]$, it follows that $(A+E)u = \lambda u$. Hence the proof. ∎

PROPOSITION 7.4. *For the* 1*-norm and the* ∞*-norm, we have*

$$\frac{|a-b|^2}{2|a-b|+4|d|} \le \mathrm{gsep} \le \mathrm{diss} \le \min\left(\frac{3}{2}\cdot\frac{|a-b|^2}{2|a-b|+4|d|},\ \frac{|a-b|^2}{4|d|}\right).$$

Obviously, if $|a-b| > |d|$, then $\frac{3}{2}\frac{|a-b|^2}{2|a-b|+4|d|}$ is smaller than $\frac{|a-b|^2}{4|d|}$. Although, diss is not known, the following example shows that the above bound provides a sharp localization of diss.

Let $A := \left[\begin{smallmatrix} 1 & 7 \\ 0 & 1.5 \end{smallmatrix}\right]$. Then $\frac{|a-b|^2}{2|a-b|+4|d|} = 8.6\times10^{-3}$ and $\frac{|a-b|^2}{4|d|} = 8.9\times10^{-3}$. Hence for the 1-norm and the ∞-norm, $8.6\times10^{-3} \le \mathrm{diss} \le 8.9\times10^{-3}$.

Next, consider $A := \left[\begin{smallmatrix} 1.1 & 0.01 \\ 0 & 1 \end{smallmatrix}\right]$. Then $\frac{|a-b|^2}{2|a-b|+4|d|} = 4.17\times10^{-2}$. This shows that $4.17\times10^{-2} \le \mathrm{diss} \le 6.25\times10^{-2}$.

Acknowledgment. The authors thank both referees for their helpful comments and suggestions, which have improved the quality of presentation.

REFERENCES

[1] R. Alam and S. Bora, *On sensitivity of eigenvalues and eigendecompositions of matrices*, Linear Algebra. Appl., 396 (2005), pp. 257–285.
[2] C. A. Bavely and G. W. Stewart, *An algorithm for computing reducing subspaces by block diagonalization*, SIAM J. Numer. Anal., 16 (1979), pp. 359–367.
[3] S. Bora, *A Geometric Analysis of Spectral Stability of Matrices and Operators*, Ph.D thesis, IIT Guwahati, India, 2001.
[4] J. W. Demmel, *Computing stable eigendecompositions of matrices*, Linear Algebra. Appl., 79 (1986), pp. 163–193.
[5] J. W. Demmel, *The condition number of equivalence transformations that block diagonalize matrix pencils*, SIAM J. Numer. Anal., 20 (1983), pp. 599–610.
[6] J. W. Demmel and B. Kågström, *Computing stable eigendecompositions of matrix pencils*, Linear Algebra. Appl., 88/89 (1987), pp. 139–186.
[7] E. Gallestey, *Computing spectral value sets using the subharmonicity of the norm of rational matrices*, BIT, 38 (1998), pp. 22–33.
[8] S. K. Godunov, *Modern Aspects of Linear Algebra*, AMS, Providence, RI, 1998.
[9] M. Gu, *Finding well-conditioned similarities to block-diagonalize non-symmetric matrices is NP-hard*, J. Complexity, 11 (1995), pp. 377–391.
[10] A. Harrabi, *Note on Pseudospectra of Closed Operators*, Technical Report TR/PA/98/08, CERFACS, Toulouse, France, 1998.
[11] B. Kågström and P. Poromaa, *Distributed and shared memory block algorithms for the triangular Sylvester equation with* sep^{-1} *estimators*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 99–101.
[12] B. Kågström and A. Rhue, *An algorithm for the numerical computation of the Jordan normal form of a complex matrix*, ACM Trans. Math. Software, 6 (1980), pp. 389–419.
[13] M. Karow, *Geometry of Spectral Value Sets*, Ph.D thesis, Universität Bremen, Germany, 2003.
[14] L. Grammont and A. Largillier, *The ϵ-spectrum and stability radius*, J. Comput. Appl. Math., 147 (2002), pp. 453–469.
[15] G. W. Stewart and J. Sun, *Matrix Perturbation Theory*, Academic Press, San Diego, CA, 1990.
[16] G. W. Stewart, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Rev., 15 (1973), pp. 727–764.
[17] L. N. Trefethen, *Pseudospectra of matrices*, in Numerical Analysis 1991, D. F. Griffiths and G. A. Watson, eds., Longman, Harlow, UK, 1992, pp. 234–266.
[18] L. N. Trefethen, *Pseudospectra of linear operators*, SIAM Rev., 39 (1997), pp. 383–406.
[19] J. M. Varah, *On the separation of two matrices*, SIAM J. Numer. Anal., 16 (1979), pp. 216–222.
[20] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, UK, 1996.
[21] J. H. Wilkinson, *Sensitivity of eigenvalues*, Util. Math., 25 (1984), pp. 5–76.
[22] J. H. Wilkinson, *Sensitivity of eigenvalues* II, Util. Math., 30 (1986), pp. 243–286.
[23] J. H. Wilkinson, *On a theorem of Feingold*, Linear Algebra. Appl., 88/89 (1987), pp. 13–30.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 3, pp. 849–877

FUNCTIONS PRESERVING MATRIX GROUPS AND ITERATIONS FOR THE MATRIX SQUARE ROOT*

NICHOLAS J. HIGHAM†, D. STEVEN MACKEY†, NILOUFER MACKEY‡, AND FRANÇOISE TISSEUR†

Abstract. For which functions f does $A \in \mathbb{G} \Rightarrow f(A) \in \mathbb{G}$ when $\mathbb{G}$ is the matrix automorphism group associated with a bilinear or sesquilinear form? For example, if A is symplectic when is $f(A)$ symplectic? We show that group structure is preserved precisely when $f(A^{-1}) = f(A)^{-1}$ for bilinear forms and when $f(A^{-*}) = f(A)^{-*}$ for sesquilinear forms. Meromorphic functions that satisfy each of these conditions are characterized. Related to structure preservation is the condition $f(\overline{A}) = \overline{f(A)}$, and analytic functions and rational functions satisfying this condition are also characterized. These results enable us to characterize all meromorphic functions that map every $\mathbb{G}$ into itself as the ratio of a polynomial and its "reversal," up to a monomial factor and conjugation.

The principal square root is an important example of a function that preserves every automorphism group $\mathbb{G}$. By exploiting the matrix sign function, a new family of coupled iterations for the matrix square root is derived. Some of these iterations preserve every $\mathbb{G}$; all of them are shown, via a novel Fréchet derivative-based analysis, to be numerically stable.

A rewritten form of Newton's method for the square root of $A \in \mathbb{G}$ is also derived. Unlike the original method, this new form has good numerical stability properties, and we argue that it is the iterative method of choice for computing $A^{1/2}$ when $A \in \mathbb{G}$. Our tools include a formula for the sign of a certain block 2×2 matrix, the generalized polar decomposition along with a wide class of iterations for computing it, and a connection between the generalized polar decomposition of $I + A$ and the square root of $A \in \mathbb{G}$.

Key words. automorphism group, bilinear form, sesquilinear form, scalar product, adjoint, Fréchet derivative, stability analysis, perplectic matrix, pseudo-orthogonal matrix, Lorentz matrix, generalized polar decomposition, matrix sign function, matrix pth root, matrix square root, structure preservation, matrix iteration, Newton iteration

AMS subject classifications. 65F30, 15A18

DOI. 10.1137/S0895479804442218

1. Introduction.

Theory and algorithms for structured matrices are of growing interest because of the many applications that generate structure and the potential benefits to be gained by exploiting it. The benefits include faster and more accurate algorithms as well as more physically meaningful solutions. Structure comes in many forms, including Hamiltonian, Toeplitz, or Vandermonde structure and total positivity. Here we study a nonlinear structure that arises in a variety of important applications and has an elegant mathematical formulation: that of a matrix automorphism group $\mathbb{G}$ associated with a bilinear or sesquilinear form.

Our particular interest is in functions that preserve matrix automorphism group structure. We show in section 3 that $A \in \mathbb{G} \Rightarrow f(A) \in \mathbb{G}$ precisely when $f(A^{-1}) =$

*Received by the editors March 18, 2004; accepted for publication (in revised form) by A. Frommer June 8, 2004; published electronically April 8, 2005.

http://www.siam.org/journals/simax/26-3/44221.html

†School of Mathematics, University of Manchester, Manchester, M13 9PL, United Kingdom (higham@ma.man.ac.uk, http://www.ma.man.ac.uk/~higham, smackey@ma.man.ac.uk, ftisseur@ma.man.ac.uk, http://www.ma.man.ac.uk/~ftisseur). The research of the first author was supported by Engineering and Physical Sciences Research Council grant GR/R22612 and by a Royal Society-Wolfson Research Merit Award. The research of the second author was supported by Engineering and Physical Sciences Research Council grant GR/S31693. The research of the fourth author was supported by Engineering and Physical Sciences Research Council grant GR/R45079.

‡Department of Mathematics, Western Michigan University, Kalamazoo, MI 49008 (nil.mackey@wmich.edu, http://homepages.wmich.edu/~mackey).

$f(A)^{-1}$ for bilinear forms or $f(A^{-*}) = f(A)^{-*}$ for sesquilinear forms; in other words, f has to commute with the inverse function or the conjugate inverse function at A. We characterize meromorphic functions satisfying each of these conditions. For sesquilinear forms, the condition $f(\overline{A}) = \overline{f(A)}$, that is, f commutes with conjugation, also plays a role in structure preservation. We characterize analytic functions and rational functions satisfying this conjugation condition. We show further that any meromorphic function that is structure preserving for all automorphism groups is rational and, up to a monomial factor and conjugation, the ratio of a polynomial and its "reversal."

The matrix sign function and the matrix principal pth root are important examples of functions that preserve all automorphism groups. Iterations for computing the sign function in a matrix group were studied by us in [15]. We concentrate here on the square root, aiming to derive iterations that exploit the group structure. Connections between the matrix sign function, the matrix square root, and the generalized polar decomposition are developed in section 4. A new identity for the matrix sign function (Lemma 4.3) establishes a link with the generalized polar decomposition (Corollary 4.4). For $A \in \mathbb{G}$ we show that the generalized polar decomposition of $I + A$ has $A^{1/2}$ as the factor in $\mathbb{G}$, thereby reducing computation of the square root to computation of the generalized polar decomposition (Theorem 4.7).

A great deal is known about iterations for the matrix sign function. Our results in section 4 show that each matrix sign function iteration of a general form leads to two further iterations:

- a coupled iteration for the principal square root of any matrix A. The iteration is structure preserving, in the sense that $A \in \mathbb{G}$ implies all the iterates lie in $\mathbb{G}$, as long as the underlying sign iteration is also structure preserving;
- an iteration for the generalized polar decomposition and hence for the square root of $A \in \mathbb{G}$.

Iterations for matrix roots are notorious for their tendency to be numerically unstable. In section 5 Fréchet derivatives are used to develop a stability analysis of the coupled square root iterations that arise from superlinearly convergent sign iterations. We find that all such iterations are stable, but that a seemingly innocuous rewriting of the iterations can make them unstable. The technique developed in this section should prove to be of wider use in analyzing matrix iterations.

In section 6 two instances of the connections identified in section 4 between the sign function and the square root are examined in detail. We obtain a family of coupled structure-preserving iterations for the square root whose members have order of convergence $2m+1$ for $m = 1, 2, \ldots$. We also derive a variant for $A \in \mathbb{G}$ of the well-known but numerically unstable Newton iteration for $A^{1/2}$ by using the connection with the generalized polar decomposition. Our numerical experiments and analysis in section 7 confirm the numerical stability of both the structure-preserving iterations and the Newton variant, showing both to be useful in practice. Because the Newton variant has a lower cost per iteration and shows better numerical preservation of structure, it is our preferred method in general.

2. Preliminaries. We give a very brief summary of the required definitions and notation. For more details, see D. S. Mackey, N. Mackey, and Tisseur [25].

Consider a scalar product on $\mathbb{K}^n$, that is, a bilinear or sesquilinear form $\langle \cdot, \cdot \rangle_M$ defined by any nonsingular matrix M: for $x, y \in \mathbb{K}^n$,

$$\langle x, y \rangle_M = \begin{cases} x^T M y & \text{for real or complex bilinear forms,} \\ x^* M y & \text{for sesquilinear forms.} \end{cases}$$

Here $\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$ and the superscript $*$ denotes conjugate transpose. The associated automorphism group is defined by

$$\mathbb{G} = \{A \in \mathbb{K}^{n\times n} : \langle Ax, Ay\rangle_{\mathrm{M}} = \langle x, y\rangle_{\mathrm{M}} \ \forall x, y \in \mathbb{K}^n\}.$$

The adjoint $A^\star$ of $A \in \mathbb{K}^{n\times n}$ with respect to $\langle\cdot,\cdot\rangle_{\mathrm{M}}$ is the unique matrix satisfying

$$\langle Ax, y\rangle_{\mathrm{M}} = \langle x, A^\star y\rangle_{\mathrm{M}} \quad \forall x, y \in \mathbb{K}^n.$$

It can be shown that the adjoint is given explicitly by

$$A^\star = \begin{cases} M^{-1}A^TM & \text{for bilinear forms,} \\ M^{-1}A^*M & \text{for sesquilinear forms} \end{cases} \tag{2.1}$$

and has the following basic properties:

$$(A+B)^\star = A^\star + B^\star, \quad (AB)^\star = B^\star A^\star, \quad (A^{-1})^\star = (A^\star)^{-1},$$
$$(\alpha A)^\star = \begin{cases} \alpha A^\star & \text{for bilinear forms,} \\ \overline{\alpha} A^\star & \text{for sesquilinear forms.} \end{cases}$$

The automorphism group can be characterized in terms of the adjoint by

$$\mathbb{G} = \{A \in \mathbb{K}^{n\times n} : A^\star = A^{-1}\}.$$

Table 2.1 lists some of the "classical" matrix groups. Observe that M, the matrix of the form, is real orthogonal with $M = \pm M^T$ in all these examples. Our results, however, place no restrictions on M other than nonsingularity; they therefore apply to all scalar products on $\mathbb{R}^n$ or $\mathbb{C}^n$ and their associated automorphism groups.

We note for later use that

$$A \in \mathbb{G} \text{ and } M \text{ unitary} \quad \Rightarrow \quad \|A\|_2 = \|A^{-1}\|_2. \tag{2.2}$$

We recall one of several equivalent ways of defining $f(A)$ for $A \in \mathbb{C}^{n\times n}$, where f is an underlying scalar function. Let A have distinct eigenvalues $\lambda_1, \ldots, \lambda_s$ occurring in Jordan blocks of maximum sizes $n_1, \ldots, n_s$, respectively. Thus if A is diagonalizable, $n_i \equiv 1$. Then $f(A) = q(A)$, where q is the unique Hermite interpolating polynomial of degree less than $\sum_{i=1}^s n_i$ that satisfies the interpolation conditions

$$q^{(j)}(\lambda_i) = f^{(j)}(\lambda_i), \quad j = 0\colon n_i - 1, \ i = 1\colon s. \tag{2.3}$$

Stated another way, q is the Hermite interpolating polynomial of minimal degree that interpolates f at the roots of the minimal polynomial of A. We use the phrase *f is defined on the spectrum of A* or, for short, *f is defined at A* or *A is in the domain of f*, to mean that the derivatives in (2.3) exist.

At various points in this work the properties $f(\mathrm{diag}(X_1, X_2)) = \mathrm{diag}(f(X_1), f(X_2))$ and $f(P^{-1}AP) = P^{-1}f(A)P$, which hold for any matrix function [24, Thms. 9.4.1, 9.4.2], will be used. We will also need the following three results.

LEMMA 2.1. *Let $A, B \in \mathbb{C}^{n\times n}$ and let f be defined on the spectrum of both A and B. Then there is a single polynomial p such that $f(A) = p(A)$ and $f(B) = p(B)$.*

Proof. It suffices to let p be the polynomial that interpolates f and its derivatives at the roots of the least common multiple of the minimal polynomials of A and B. See the discussion in [16, p. 415]. ∎

TABLE 2.1
A sampling of automorphism groups.

Here, $R = \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix}$, $J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$, $\Sigma_{p,q} = \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} \in \mathbb{R}^{n\times n}$.

Space	M	$A^\star$	Automorphism group, $\mathbb{G}$
Groups corresponding to a bilinear form			
$\mathbb{R}^n$	I	$A^\star = A^T$	Real orthogonals
$\mathbb{C}^n$	I	$A^\star = A^T$	Complex orthogonals
$\mathbb{R}^n$	$\Sigma_{p,q}$	$A^\star = \Sigma_{p,q}A^T\Sigma_{p,q}$	Pseudo-orthogonals[a]
$\mathbb{R}^n$	R	$A^\star = RA^TR$	Real perplectics
$\mathbb{R}^{2n}$	J	$A^\star = -JA^TJ$	Real symplectics
$\mathbb{C}^{2n}$	J	$A^\star = -JA^TJ$	Complex symplectics
Groups corresponding to a sesquilinear form			
$\mathbb{C}^n$	I	$A^\star = A^*$	Unitaries
$\mathbb{C}^n$	$\Sigma_{p,q}$	$A^\star = \Sigma_{p,q}A^*\Sigma_{p,q}$	Pseudo-unitaries
$\mathbb{C}^{2n}$	J	$A^\star = -JA^*J$	Conjugate symplectics

[a] Also known as Lorentz matrices.

COROLLARY 2.2. *Let $A, B \in \mathbb{C}^{n\times n}$ and let f be defined on the spectra of both AB and BA. Then*

$$Af(BA) = f(AB)A.$$

Proof. By Lemma 2.1 there is a single polynomial p such that $f(AB) = p(AB)$ and $f(BA) = p(BA)$. Hence

$$Af(BA) = Ap(BA) = p(AB)A = f(AB)A. \qquad \square$$

LEMMA 2.3. *Any rational function r can be uniquely represented in the form $r(z) = z^n p(z)/q(z)$, where p is monic, n is an integer, p and q are relatively prime, and $p(0)$ and $q(0)$ are both nonzero.*

Proof. The proof of the lemma is straightforward. $\square$

We denote the closed negative real axis by $\mathbb{R}^-$. For $A \in \mathbb{C}^{n\times n}$ with no eigenvalues on $\mathbb{R}^-$, the principal matrix pth root $A^{1/p}$ is defined by the property that the eigenvalues of $A^{1/p}$ lie in the segment $\{z : -\pi/p < \arg(z) < \pi/p\}$. We will most often use the principal square root, $A^{1/2}$, whose eigenvalues lie in the open right half-plane.

Finally, we introduce some notation connected with a polynomial p. The polynomial obtained by replacing the coefficients of p by their conjugates is denoted by $\overline{p}$. The polynomial obtained by reversing the order of the coefficients of p is denoted by $\mathrm{rev}p$; thus if p has degree m then

$$\mathrm{rev}p(x) = x^m p(1/x). \tag{2.4}$$

3. Structure-preserving functions. Our aim in this section is to characterize functions f that preserve automorphism group structure. For a given $\mathbb{G}$, if $f(A) \in \mathbb{G}$ for all $A \in \mathbb{G}$ for which $f(A)$ is defined, we will say that f *is structure preserving for* $\mathbb{G}$. As well as determining f that preserve structure for a particular $\mathbb{G}$, we wish to determine f that preserve structure *for all* $\mathbb{G}$.

The (principal) square root is an important example of a function that preserves all groups. To see this for $\mathbb{G}$ associated with a bilinear form, recall that $A \in \mathbb{G}$ is equivalent to $M^{-1}A^TM = A^{-1}$. Assuming that A has no eigenvalues on $\mathbb{R}^-$, taking the (principal) square root in this relation gives

$$A^{-1/2} = \bigl(M^{-1}A^TM\bigr)^{1/2} = M^{-1}\bigl(A^T\bigr)^{1/2}M = M^{-1}\bigl(A^{1/2}\bigr)^TM, \tag{3.1}$$

which shows that $A^{1/2} \in \mathbb{G}$.

In order to understand structure preservation we first need to characterize when $f(A) \in \mathbb{G}$ for a fixed f and a fixed $A \in \mathbb{G}$. The next result relates this property to various other relevant properties of matrix functions.

THEOREM 3.1. *Let $\mathbb{G}$ be the automorphism group of a scalar product. Consider the following eight properties of a matrix function f at a (fixed) matrix $A \in \mathbb{K}^{n\times n}$, where f is assumed to be defined at the indicated arguments:*

(a) $f(A^T) = f(A)^T$,
(b) $f(A^*) = f(A)^*$,
(c) $f(\overline{A}) = \overline{f(A)}$,
(d) $f(A^\star) = f(A)^\star$,
(e) $f(A^{-1}) = f(A)^{-1}$,
(g) $f(A^{-*}) = f(A)^{-*}$,
(h) $f(A^{-\star}) = f(A)^{-\star}$,
(i) *when* $A \in \mathbb{G}$, $f(A) \in \mathbb{G}$.

(a) *always holds.* (b) *is equivalent to* (c). (c) *is equivalent to the existence of a single* real *polynomial p such that $f(A) = p(A)$ and $f(\overline{A}) = p(\overline{A})$. Moreover,*

- for bilinear forms: (d) *always holds.* (e), (h), *and* (i) *are equivalent;*
- for sesquilinear forms: (d) *is equivalent to* (b) *and to* (c). (g), (h), *and* (i) *are equivalent. Any two of* (d) *for A^{-1} and* (e) *and* (h) *for A imply the third.*[1]

Proof. (a) follows because the same polynomial can be used to evaluate $f(A)$ and $f(A^T)$, by Lemma 2.1. Property (b) is equivalent to $f(\overline{A}^T) = \overline{f(A)}^T$, which on applying (a) becomes (c). So (b) is equivalent to (c).

Next, we consider the characterization of (c). Suppose p is a real polynomial such that $f(A) = p(A)$ and $f(\overline{A}) = p(\overline{A})$. Then $f(\overline{A}) = p(\overline{A}) = \overline{p(A)} = \overline{f(A)}$, which is (c). Conversely, assume (c) holds and let q be any complex polynomial that simultaneously evaluates f at A and $\overline{A}$, so that $f(A) = q(A)$ and $f(\overline{A}) = q(\overline{A})$; the existence of such a q is assured by Lemma 2.1. Then

$$q(A) = f(A) = \overline{f(\overline{A})} = \overline{q(\overline{A})} = \overline{q}(A),$$

and hence $p(x) := \frac{1}{2}(q(x) + \overline{q}(x))$ is a *real* polynomial such that $f(A) = p(A)$. Since

$$q(\overline{A}) = f(\overline{A}) = \overline{f(A)} = \overline{q(A)} = \overline{q}(\overline{A}),$$

we also have $f(\overline{A}) = p(\overline{A})$.

We now consider (d). From the characterization (2.1) of the adjoint, for bilinear forms we have

$$f(A^\star) = f(M^{-1}A^TM) = M^{-1}f(A^T)M = M^{-1}f(A)^TM = f(A)^\star,$$

so (d) always holds. For sesquilinear forms,

$$f(A^\star) = f(M^{-1}A^*M) = M^{-1}f(A^*)M,$$

[1] We will see at the end of section 3.4 that for sesquilinear forms neither (d) for A^{-1} (equivalently (c) for A^{-1}) nor (e) is a necessary condition for (h) (or, equivalently, for the structure-preservation property (i)).

which equals $f(A)^\star = M^{-1}f(A)^*M$ if and only if (b) holds.

To see that (h) and (i) are equivalent, consider the following cycle of potential equalities:

$$\begin{array}{ccc} f(A^{-\star}) & \overset{\text{(h)}}{=\!=} & f(A)^{-\star} \\ {\scriptstyle \text{(x)}}\ \backslash\backslash & & //\ {\scriptstyle \text{(y)}} \\ & f(A) & \end{array}$$

Clearly (x) holds if $A \in \mathbb{G}$, and (y) holds when $f(A) \in \mathbb{G}$. Hence (h) and (i) are equivalent.

For bilinear forms,

$$\begin{aligned} f(A^{-\star}) = f(A)^{-\star} &\iff f(M^{-1}A^{-T}M) = M^{-1}f(A)^{-T}M \\ &\iff f(A^{-T}) = f(A)^{-T} \\ &\iff f(A^{-1}) = f(A)^{-1} \quad \text{by (a).} \end{aligned}$$

Thus (h) is equivalent to (e). For sesquilinear forms a similar argument shows that (h) is equivalent to (g).

Finally, it is straightforward to show for sesquilinear forms that any two of (d) for A^{-1} and (e) and (h) for A imply the third. ∎

The main conclusion of Theorem 3.1 is that f is structure preserving for $\mathbb{G}$ precisely when $f(A^{-1}) = f(A)^{-1}$ for all $A \in \mathbb{G}$ for bilinear forms, or $f(A^{-*}) = f(A)^{-*}$ for all $A \in \mathbb{G}$ for sesquilinear forms. We can readily identify two important functions that satisfy both these conditions more generally for all $A \in \mathbb{K}^{n\times n}$ in their domains, and hence are structure preserving *for all* $\mathbb{G}$.

- The matrix sign function. Recall that for a matrix $A \in \mathbb{C}^{n\times n}$ with no pure imaginary eigenvalues the sign function can be defined by $\operatorname{sign}(A) = A(A^2)^{-1/2}$ [12], [22]. That the sign function is structure preserving is known: proofs specific to the sign function are given in [15] and [26].
- Any matrix power A^α, subject for fractional α to suitable choice of the branches of the power at each eigenvalue; in particular, the principal matrix pth root $A^{1/p}$. The structure-preserving property of the principal *square* root is also shown by D. S. Mackey, N. Mackey, and Tisseur [26].

In the following three subsections we investigate three of the properties in Theorem 3.1 in detail, for general matrices $A \in \mathbb{C}^{n\times n}$. Then in the final two subsections we characterize meromorphic structure-preserving functions and conclude with a brief consideration of M-normal matrices.

3.1. Property (c): $f(\overline{A}) = \overline{f(A)}$. Theorem 3.1 shows that this property for A^{-1}, together with property (e), namely, $f(A^{-1}) = f(A)^{-1}$, is sufficient for structure preservation in the sesquilinear case. While property (c) is not necessary for structure preservation, it plays an important role in our understanding of the preservation of realness, and so is of independent interest.

We first give a characterization of analytic functions satisfying property (c) for all A in their domain, followed by an explicit description of all rational functions with the property. We denote by $\Lambda(A)$ the set of eigenvalues of A.

THEOREM 3.2. *Let f be analytic on an open subset $\Omega \subseteq \mathbb{C}$ such that each connected component of Ω is closed under conjugation. Consider the corresponding matrix function f on its natural domain in $\mathbb{C}^{n\times n}$, the set $\mathcal{D} = \{A \in \mathbb{C}^{n\times n} : \Lambda(A) \subseteq \Omega\}$. Then the following are equivalent:*

(a) $f(\overline{A}) = \overline{f(A)}$ *for all* $A \in \mathcal{D}$.
(b) $f(\mathbb{R}^{n\times n} \cap \mathcal{D}) \subseteq \mathbb{R}^{n\times n}$.
(c) $f(\mathbb{R} \cap \Omega) \subseteq \mathbb{R}$.

Proof. Our strategy is to show that (a) $\Rightarrow$ (b) $\Rightarrow$ (c) $\Rightarrow$ (a).
(a) $\Rightarrow$ (b): If $A \in \mathbb{R}^{n\times n} \cap \mathcal{D}$ then

$$\begin{aligned} f(A) &= f(\overline{A}) \quad (\text{since } A \in \mathbb{R}^{n\times n}) \\ &= \overline{f(A)} \quad (\text{given}), \end{aligned}$$

so $f(A) \in \mathbb{R}^{n\times n}$, as required.

(b) $\Rightarrow$ (c): If $\lambda \in \mathbb{R} \cap \Omega$ then $\lambda I \in \mathcal{D}$. But $f(\lambda I) \in \mathbb{R}^{n\times n}$ by (b), and hence, since $f(\lambda I) = f(\lambda)I$, $f(\lambda) \in \mathbb{R}$.

(c) $\Rightarrow$ (a): Let $\widetilde{\Omega}$ be any connected component of Ω. Since $\widetilde{\Omega}$ is open and connected it is path-connected, and since it is also closed under conjugation it must contain some $\lambda \in \mathbb{R}$ by the intermediate value theorem. The openness of $\widetilde{\Omega}$ in $\mathbb{C}$ then implies that $U = \widetilde{\Omega} \cap \mathbb{R}$ is a nonempty open subset of $\mathbb{R}$, with $f(U) \subseteq \mathbb{R}$ by hypothesis. Now since f is analytic on $\widetilde{\Omega}$, it follows from the "identity theorem" [27, pp. 227–236 and Ex. 4, p. 236] that $f(\overline{z}) = \overline{f(z)}$ for all $z \in \widetilde{\Omega}$. The same argument applies to all the other connected components of Ω, so $f(\overline{z}) = \overline{f(z)}$ for all $z \in \Omega$. Thus $f(\overline{A}) = \overline{f(A)}$ holds for all diagonal matrices in $\mathcal{D}$, and hence for all diagonalizable matrices in $\mathcal{D}$. Since the scalar function f is analytic on Ω, the matrix function f is continuous on $\mathcal{D}$ [16, Thm. 6.2.27][2] and therefore the identity holds for all matrices in $\mathcal{D}$, since diagonalizable matrices are dense in any open subset of $\mathbb{C}^{n\times n}$. ∎

Turning to the case when f is rational, we need a preliminary lemma.

LEMMA 3.3. *Suppose r is a complex rational function that maps all reals (in its domain) to reals. Then r can be expressed as the ratio of two real polynomials. In particular, in the canonical form for r given by Lemma* 2.3 *the polynomials p and q are both real.*

Proof. Let $r(z) = z^n p(z)/q(z)$ be the canonical form of Lemma 2.3 and consider the rational function

$$h(z) := \overline{r(\overline{z})} = z^n \frac{\overline{p}(z)}{\overline{q}(z)}.$$

Clearly, $h(z) = r(z)$ for all real z in the domain of r, and hence $p(z)/q(z) = \overline{p}(z)/\overline{q}(z)$ for this infinitude of z. It is then straightforward to show (cf. the proof of Lemma 3.6 below) that $p = \alpha\overline{p}$ and $q = \alpha\overline{q}$ for some nonzero $\alpha \in \mathbb{C}$. But the monicity of p implies that $\alpha = 1$, so p and q are real polynomials. ∎

Combining Lemma 3.3 with Theorem 3.2 gives a characterization of all rational matrix functions with property (c) in Theorem 3.1.

THEOREM 3.4. *A rational matrix function $r(A)$ has the property $r(\overline{A}) = \overline{r(A)}$ for all $A \in \mathbb{C}^{n\times n}$ such that A and $\overline{A}$ are in the domain of r if and only if the scalar function r can be expressed as the ratio of two real polynomials. In particular, the polynomials p satisfying $p(\overline{A}) = \overline{p(A)}$ for all $A \in \mathbb{C}^{n\times n}$ are precisely those with real coefficients.*

[2] Horn and Johnson require that Ω should be a *simply-connected* open subset of $\mathbb{C}$. However, it is not difficult to show that just the openness of Ω is sufficient to conclude that the matrix function f is continuous on $\mathcal{D}$.

3.2. Property (e): $f(A^{-1}) = f(A)^{-1}$. We now investigate further the property $f(A^{-1}) = f(A)^{-1}$ for matrix functions f. We would like to know when this property holds for all A such that A and A^{-1} are in the domain of f. Since a function of a diagonal matrix is diagonal, a necessary condition on f is that $f(z)f(1/z) = 1$ whenever z and $1/z$ are in the domain of f. The following result characterizes meromorphic functions satisfying this identity. Recall that a function is said to be meromorphic on an open subset $U \subseteq \mathbb{C}$ if it is analytic on U except for poles. In this paper we consider only meromorphic functions on $\mathbb{C}$, so the phrase "f is meromorphic" will mean f is meromorphic on $\mathbb{C}$.

LEMMA 3.5. *Suppose f is a meromorphic function on $\mathbb{C}$ such that $f(z)f(1/z) = 1$ holds for all z in some infinite compact subset of $\mathbb{C}$. Then*

(a) *The identity $f(z)f(1/z) = 1$ holds for all nonzero $z \in \mathbb{C} \setminus S$, where S is the discrete set consisting of the zeros and poles of f together with their reciprocals.*

(b) *The zeros and poles of f come in reciprocal pairs $\{a, 1/a\}$ with matching orders. That is,*

$$z = a \text{ is a zero (pole) of order } k \iff z = 1/a \text{ is a pole (zero) of order } k. \tag{3.2}$$

Consequently, the set S is finite and consists of just the zeros and poles of f. Note that $\{0, \infty\}$ is also to be regarded as a reciprocal pair for the purpose of statement (3.2).

(c) *The function f is meromorphic at ∞.*

(d) *The function f is rational.*

Proof. (a) The function $g(z) := f(z)f(1/z)$ is analytic on the open connected set $\mathbb{C} \setminus \{S \cup \{0\}\}$, so the result follows by the identity theorem.

(b) Consider first the case where $a \neq 0$ is a zero or a pole of f. Because f is meromorphic the set S is discrete, so by (a) there is some open neighborhood U of $z = a$ such that $f(z)f(1/z) = 1$ holds for all $z \in U \setminus \{a\}$ *and* such that f can be expressed as $f(z) = (z-a)^k g(z)$ for some nonzero $k \in \mathbb{Z}$ ($k > 0$ for a zero, $k < 0$ for a pole) and some function g that is analytic and nonzero on all of U. Then for all $z \in U \setminus \{a\}$ we have

$$f(1/z) = \frac{1}{f(z)} = (z-a)^{-k} \frac{1}{g(z)}.$$

Letting $w = 1/z$, we see that there is an open neighborhood $\widetilde{U}$ of $w = 1/a$ in which

$$f(w) = \left(\frac{1}{w} - a\right)^{-k} \frac{1}{g(1/w)} = \left(w - \frac{1}{a}\right)^{-k} \frac{(-1)^k w^k}{a^k g(1/w)} =: \left(w - \frac{1}{a}\right)^{-k} h(w)$$

holds for all $w \in \widetilde{U} \setminus \{\frac{1}{a}\}$, where $h(w)$ is analytic and nonzero for all $w \in \widetilde{U}$. This establishes (3.2) and hence that the set S consists of just the zeros and poles of f.

Next we turn to the case of the "reciprocal" pair $\{0, \infty\}$. First note that the zeros and poles of any nonzero meromorphic function can never accumulate at any finite point z, so in particular $z = 0$ cannot be a limit point of S. In our situation the set S also cannot have $z = \infty$ as an accumulation point; if it did, then the reciprocal pairing of the nonzero poles and zeros of f just established would force $z = 0$ to be a limit point of S. Thus if $z = \infty$ is a zero or singularity of f then it must be an isolated zero or singularity, which implies that S is a finite set.

Now suppose $z = 0$ is a zero or pole of f. In some open neighborhood U of $z = 0$ we can write $f(z) = z^k g(z)$ for some nonzero $k \in \mathbb{Z}$ and some g that is analytic and

nonzero on U. Then for all $z \in U \setminus \{0\}$ we have

$$f(1/z) = \frac{1}{f(z)} = z^{-k}\frac{1}{g(z)} =: z^{-k}h(z),$$

where h is analytic and nonzero in U. Thus $z = 0$ being a zero (pole) of f implies that $z = \infty$ is a pole (zero) of f. The converse is established by the same kind of argument.

(c) That f is meromorphic at ∞ follows from (3.2), the finiteness of S, and the identity $f(z)f(1/z) = 1$, together with the fact that f (being meromorphic on $\mathbb{C}$) can have only a pole, a zero, or a finite value at $z = 0$.

(d) By [9, Thm. 4.7.7] a function is meromorphic on $\mathbb{C}$ and at ∞ if and only if it is rational. ∎

Since Lemma 3.5 focuses attention on rational functions, we next give a complete description of all rational functions satisfying the identity $f(z)f(1/z) = 1$. Recall that $\mathrm{rev}p$ is defined by (2.4).

LEMMA 3.6. *A complex rational function $r(z)$ satisfies the identity $r(z)r(1/z) = 1$ for infinitely many $z \in \mathbb{C}$ if and only if it can be expressed in the form*

$$r(z) = \pm z^k \frac{p(z)}{\mathrm{rev}p(z)} \tag{3.3}$$

for some $k \in \mathbb{Z}$ and some polynomial p. For any r of the form (3.3) *the identity $r(z)r(1/z) = 1$ holds for all nonzero $z \in \mathbb{C}$ except for the zeros of p and their reciprocals. Furthermore, there is always a unique choice of p in* (3.3) *so that p is monic, p and $\mathrm{rev}p$ are relatively prime, and $p(0) \neq 0$; in this case the sign is also uniquely determined. In addition, $r(z)$ is real whenever z is real if and only if this unique p is real.*

Proof. For any r of the form (3.3) it is easy to check that $r(1/z) = \pm z^{-k}(\mathrm{rev}p(z))/p(z)$, so that the identity $r(z)r(1/z) = 1$ clearly holds for all nonzero $z \in \mathbb{C}$ except for the zeros of p and their reciprocals (which are the zeros of $\mathrm{rev}p$).

Conversely, suppose that $r(z)$ satisfies $r(z)r(1/z) = 1$ for infinitely many $z \in \mathbb{C}$. By Lemma 2.3, we can uniquely write r as $r(z) = z^k p(z)/q(z)$, where p and q are relatively prime, p is monic, and $p(0)$ and $q(0)$ are both nonzero. For this unique representation of r, we will show that $q(z) = \pm\mathrm{rev}p(z)$, giving us the form (3.3). Begin by rewriting the condition $r(z)r(1/z) = 1$ as

$$p(z)p(1/z) = q(z)q(1/z). \tag{3.4}$$

Letting n be any integer larger than $\deg p$ and $\deg q$ (where $\deg p$ denotes the degree of p), multiplying both sides of (3.4) by z^n and using the definition of rev results in

$$z^{n-\deg p}p(z)\,\mathrm{rev}p(z) = z^{n-\deg q}q(z)\,\mathrm{rev}q(z).$$

Since this equality of polynomials holds for infinitely many z, it must be an identity. Thus $\deg p = \deg q$, and

$$p(z)\,\mathrm{rev}p(z) = q(z)\,\mathrm{rev}q(z) \tag{3.5}$$

holds for all $z \in \mathbb{C}$. Since p has no factors in common with q, p must divide $\mathrm{rev}q$. Therefore $\mathrm{rev}q(z) = \alpha p(z)$ for some $\alpha \in \mathbb{C}$, which implies $q(z) = \alpha\,\mathrm{rev}p(z)$ since $q(0) \neq 0$. Substituting into (3.5) gives $\alpha^2 = 1$, so that $q(z) = \pm\mathrm{rev}p(z)$, as desired. The final claim follows from Lemma 3.3. ∎

Lemmas 3.5 and 3.6 now lead us to the following characterization of meromorphic matrix functions with the property $f(A^{-1}) = f(A)^{-1}$. Here and in the rest of this paper we use the phrase "meromorphic matrix function $f(A)$" to mean a matrix function whose underlying scalar function f is meromorphic on all of $\mathbb{C}$.

THEOREM 3.7. *A meromorphic matrix function $f(A)$ has the property $f(A^{-1}) = f(A)^{-1}$ for all $A \in \mathbb{C}^{n\times n}$ such that A and A^{-1} are in the domain of f if and only if the scalar function f is rational and can be expressed in the form*

$$f(z) = \pm z^k \frac{p(z)}{\mathrm{rev}p(z)} \tag{3.6}$$

for some $k \in \mathbb{Z}$ and some polynomial p. If desired, the polynomial p may be chosen (uniquely) so that p is monic, p and $\mathrm{rev}p$ are relatively prime, and $p(0) \neq 0$. The matrix function $f(A)$ maps real matrices to real matrices if and only if this unique p is real.

Proof. As noted at the start of this subsection, $f(z)f(1/z) = 1$ for all z such that z and $1/z$ are in the domain of f is a necessary condition for having the property $f(A^{-1}) = f(A)^{-1}$. That f is rational then follows from Lemma 3.5, and from Lemma 3.6 we see that f must be of the form (3.6). To prove sufficiency, consider any f of the form (3.6) with $\deg p = n$. Then we have

$$\begin{aligned} f(A)f(A^{-1}) &= \pm A^k p(A)[\mathrm{rev}p(A)]^{-1} \cdot \pm A^{-k} p(A^{-1})[\mathrm{rev}p(A^{-1})]^{-1} \\ &= A^k p(A)[A^n p(A^{-1})]^{-1} \cdot A^{-k} p(A^{-1})[A^{-n} p(A)]^{-1} \\ &= A^k p(A) p(A^{-1})^{-1} A^{-n} \cdot A^{-k} p(A^{-1}) p(A)^{-1} A^n \\ &= I. \end{aligned}$$

The final claim follows from Theorem 3.2 and Lemma 3.6. ∎

3.3. Property (g): $f(A^{-*}) = f(A)^{-*}$. The results in this section provide a characterization of meromorphic matrix functions satisfying $f(A^{-*}) = f(A)^{-*}$. Consideration of the action of f on diagonal matrices leads to the identity $f(z)\overline{f(1/\overline{z})} = 1$ as a necessary condition on f. Thus the analysis of this identity is a prerequisite for understanding the corresponding matrix function property.

The following analogue of Lemma 3.5 can be derived, with a similar proof.

LEMMA 3.8. *Suppose f is a meromorphic function on $\mathbb{C}$ such that $f(z)\overline{f(1/\overline{z})} = 1$ holds for all z in some infinite compact subset of $\mathbb{C}$. Then f is a rational function with its zeros and poles matched in conjugate reciprocal pairs $\{a, 1/\overline{a}\}$. That is,*

$$z = a \text{ is a zero (pole) of order } k \iff z = 1/\overline{a} \text{ is a pole (zero) of order } k, \tag{3.7}$$

where $\{0, \infty\}$ is also to be regarded as a conjugate reciprocal pair.

In view of this result we can restrict our attention to rational functions.

LEMMA 3.9. *A complex rational function $r(z)$ satisfies the identity $r(z)\overline{r(1/\overline{z})} = 1$ for infinitely many $z \in \mathbb{C}$ if and only if it can be expressed in the form*

$$r(z) = \alpha z^k \frac{p(z)}{\mathrm{rev}\overline{p}(z)} \tag{3.8}$$

for some $k \in \mathbb{Z}$, some $|\alpha| = 1$, and some polynomial p. For any r of the form (3.8) the identity $r(z)\overline{r(1/\overline{z})} = 1$ holds for all nonzero $z \in \mathbb{C}$ except for the zeros of p and their conjugate reciprocals. Furthermore, there is always a unique choice of p in (3.8)

so that p is monic, p and $\mathrm{rev}\overline{p}$ are relatively prime, and $p(0) \neq 0$; in this case the scalar α is also unique. In addition, $r(z)$ is real whenever z is real if and only if this unique p is real and $\alpha = \pm 1$.

Proof. The proof is entirely analogous to that of Lemma 3.6 and so is omitted. ∎

With Lemmas 3.8 and 3.9 we can now establish the following characterization of meromorphic functions with the property $f(A^{-*}) = f(A)^{-*}$.

THEOREM 3.10. *A meromorphic matrix function $f(A)$ has the property $f(A^{-*}) = f(A)^{-*}$ for all $A \in \mathbb{C}^{n\times n}$ such that A and A^{-*} are in the domain of f if and only if the scalar function f is rational and can be expressed in the form*

$$f(z) = \alpha z^k \frac{p(z)}{\mathrm{rev}\overline{p}(z)} \tag{3.9}$$

for some $k \in \mathbb{Z}$, some $|\alpha| = 1$, and some polynomial p. If desired, the polynomial p may be chosen (uniquely) so that p is monic, p and $\mathrm{rev}\overline{p}$ are relatively prime, and $p(0) \neq 0$; in this case the scalar α is also unique. The matrix function $f(A)$ maps real matrices to real matrices if and only if this unique p is real and $\alpha = \pm 1$.

Proof. As noted at the start of this subsection, $f(z)\overline{f(1/\overline{z})} = 1$ for all z such that z and $1/\overline{z}$ are in the domain of f is a necessary condition for having the property $f(A^{-*}) = f(A)^{-*}$. That f is rational then follows from Lemma 3.8, and from Lemma 3.9 we see that f must be of the form (3.9). To prove sufficiency, consider any f of the form (3.9) with $\deg p = n$. Then we have

$$\begin{aligned}
f(A)^* f(A^{-*}) &= \left[\alpha A^k p(A)[\mathrm{rev}\overline{p}(A)]^{-1}\right]^* \cdot \alpha(A^{-*})^k p(A^{-*})[\mathrm{rev}\overline{p}(A^{-*})]^{-1} \\
&= \left[\alpha A^k p(A)[A^n \overline{p}(A^{-1})]^{-1}\right]^* \cdot \alpha(A^*)^{-k} p(A^{-*})[(A^{-*})^n \overline{p}(A^*)]^{-1} \\
&= (A^*)^{-n}\overline{p}(A^{-1})^{-*} p(A)^* (A^*)^k \overline{\alpha}\alpha (A^*)^{-k} p(A^{-*})\overline{p}(A^*)^{-1}(A^*)^n \\
&= (A^*)^{-n} p(A^{-*})^{-1}\overline{p}(A^*) p(A^{-*})\overline{p}(A^*)^{-1}(A^*)^n \\
&= I.
\end{aligned}$$

The final claim follows from Lemma 3.9. ∎

Perhaps surprisingly, one can also characterize *general* analytic functions f satisfying $f(A^{-*}) = f(A)^{-*}$. The next result has a proof very similar to that of Theorem 3.2.

THEOREM 3.11. *Let f be analytic on an open subset $\Omega \subseteq \mathbb{C}$ such that each connected component of Ω is closed under reciprocal conjugation (i.e., under the map $z \mapsto 1/\overline{z}$). Consider the corresponding matrix function f on its natural domain in $\mathbb{C}^{n\times n}$, the set $\mathcal{D} = \{A \in \mathbb{C}^{n\times n} : \Lambda(A) \subseteq \Omega\}$. Then the following are equivalent:*

(a) *$f(A^{-*}) = f(A)^{-*}$ for all $A \in \mathcal{D}$.*
(b) *$f(U(n)\cap\mathcal{D}) \subseteq U(n)$, where $U(n)$ denotes the group of $n\times n$ unitary matrices.*
(c) *$f(C \cap \Omega) \subseteq C$, where C denotes the unit circle $\{z : |z| = 1\}$.*

This theorem has the striking corollary that if a function is structure preserving for the unitary group then it is automatically structure preserving for *any other* automorphism group associated with a sesquilinear form.

COROLLARY 3.12. *Consider any function f satisfying the conditions of Theorem 3.11. Then f is structure preserving for all $\mathbb{G}$ associated with a sesquilinear form if and only if f is structure preserving for the unitary group $U(n)$.*

In view of the connection between the identity $f(z)\overline{f(1/\overline{z})} = 1$ and the property $f(C) \subseteq C$ established by Theorem 3.11, we can now see Lemma 3.9 as a natural

generalization of the well-known classification of all Möbius transformations mapping the open unit disc bijectively to itself, and hence mapping the unit circle to itself. These transformations are given by [9, Thm. 6.2.3], [27, sect. 2.3.3]

$$f(z) = \alpha \frac{z - \beta}{1 - \overline{\beta} z},$$

where α and β are any complex constants satisfying $|\alpha| = 1$ and $|\beta| < 1$. This formula is easily seen to be a special case of Lemma 3.9.

3.4. Structure-preserving meromorphic functions. We can now give a complete characterization of structure-preserving meromorphic functions. This result extends [15, Thm. 2.1], which covers the "if" case in part (e).

THEOREM 3.13. *Consider the following two types of rational functions, where* $k \in \mathbb{Z}$, $|\alpha| = 1$, *and* p *is a polynomial:*

$$\text{(I)}: \ \pm z^k \frac{p(z)}{\mathrm{rev} p(z)}, \qquad \text{(II)}: \ \alpha z^k \frac{p(z)}{\mathrm{rev}\overline{p}(z)}.$$

Let $\mathbb{G}$ *denote the automorphism group of a scalar product. A meromorphic matrix function* f *is structure preserving for all groups* $\mathbb{G}$ *associated with*

(a) *a bilinear form on* $\mathbb{C}^n$ *if and only if* f *can be expressed in Type* I *form;*

(b) *a bilinear form on* $\mathbb{R}^n$ *if and only if* f *can be expressed in Type* I *form with a* real p*;*

(c) *a sesquilinear form on* $\mathbb{C}^n$ *if and only if* f *can be expressed in Type* II *form;*

(d) *a scalar product on* $\mathbb{C}^n$ *if and only if* f *can be expressed in Type* I *form with a* real p*;*

(e) *any scalar product if and only if* f *can be expressed in Type* I *form with a* real p.

Any such structure-preserving function can be uniquely expressed with a monic polynomial p *such that* p *and* $\mathrm{rev}p$ (*or* p *and* $\mathrm{rev}\overline{p}$ *for Type* II) *are relatively prime and* $p(0) \neq 0$.

Proof. (a) Theorem 3.1 shows that structure preservation is equivalent to the condition $f(A^{-1}) = f(A)^{-1}$ for all $A \in \mathbb{G}$ (in the domain of f), although not necessarily for all $A \in \mathbb{C}^{n\times n}$ (in the domain of f). Thus we cannot directly invoke Theorem 3.7 to reach the desired conclusion. However, note that the complex symplectic group contains diagonal matrices with arbitrary nonzero complex numbers z in the (1,1) entry. Thus $f(z)f(1/z) = 1$ for all nonzero complex numbers in the domain of f is a necessary condition for $f(A^{-1}) = f(A)^{-1}$ to hold for all $\mathbb{G}$. Hence f must be rational by Lemma 3.5, and must be a Type I rational by Lemma 3.6. That being of Type I is sufficient for structure preservation follows from Theorem 3.7.

(b) The argument used in part (a) also proves (b), simply by replacing the word "complex" throughout by "real" and noting that Lemma 3.6 implies that p may be chosen to be real.

(c) The argument of part (a) can be adapted to the sesquilinear case. By Theorem 3.1, structure preservation is in this case equivalent to the condition $f(A^{-*}) = f(A)^{-*}$ for all $A \in \mathbb{G}$ (in the domain of f). Again we cannot directly invoke Theorem 3.10 to complete the argument, but a short detour through Lemmas 3.8 and 3.9 will yield the desired conclusion. Observe that the *conjugate* symplectic group contains diagonal matrices D with arbitrary nonzero complex numbers z in the (1,1) entry. The condition $f(D^{-*}) = f(D)^{-*}$ then implies that $f(1/\overline{z}) = 1/\overline{f(z)}$ for all

nonzero z in the domain of f, or, equivalently, $f(z)\overline{f(1/\overline{z})} = 1$. Lemma 3.8 now implies that f must be rational, Lemma 3.9 implies that f must be of Type II, and Theorem 3.10 then shows that any Type II rational function is indeed structure preserving.

(d) The groups considered here are the union of those in (a) and (c), so any structure-preserving f can be expressed in both Type I and Type II forms. But Lemmas 3.6 and 3.9 show that when f is expressed in the Lemma 2.3 canonical form $z^n p(z)/q(z)$, with p monic, $p(0) \neq 0$, and p relatively prime to q, then this particular expression for f is simultaneously of Type I and Type II. Thus $q = \pm \mathrm{rev} p = \beta \mathrm{rev} \overline{p}$ for some $|\beta| = 1$, and hence $p = \gamma \overline{p}$ ($\gamma = \pm\beta$). The monicity of p then implies $\gamma = 1$, so that p must be real. Conversely, it is clear that any Type I rational with real p is also of Type II and hence is structure preserving for automorphism groups of both bilinear and sesquilinear forms on $\mathbb{C}^n$.

(e) Finally, (b) and (d) together immediately imply (e). ∎

We note a subtle feature of the sesquilinear case. That the conditions $f(A^{-1}) = f(A)^{-1}$ and $f(\overline{A}) = \overline{f(A)}$ hold for all $A \in \mathbb{G}$ is sufficient for f to be structure preserving for $\mathbb{G}$ (as shown by Theorem 3.1). However, neither condition is necessary, as a simple example shows. Consider the function $f(z) = iz$. Since f is a Type II rational, it is structure preserving by Theorem 3.13(c). But it is easy to see that neither $f(A^{-1}) = f(A)^{-1}$ nor $f(\overline{A}) = \overline{f(A)}$ holds for any nonzero matrix A.

3.5. M-normal matrices. We conclude this section with a structure-preservation result of a different flavor. It is well known that if A is normal ($A^*A = AA^*$) then $f(A)$ is normal. This result can be generalized to an arbitrary scalar product space. Following Gohberg, Lancaster, and Rodman [8, sect. I.4.6], define $A \in \mathbb{K}^{n\times n}$ to be M-normal, that is, normal with respect to the scalar product, defined by a matrix M, if $A^\star A = AA^\star$. If A belongs to the automorphism group $\mathbb{G}$ of the scalar product, then A is certainly M-normal. The next result shows that any function f preserves M-normality. In particular, for $A \in \mathbb{G}$, even though $f(A)$ may not belong to $\mathbb{G}$, $f(A)$ is M-normal and so has some structure.

THEOREM 3.14. *Consider a scalar product defined by a matrix M. Let $A \in \mathbb{K}^{n\times n}$ be M-normal and let f be any function defined on the spectrum of A. Then $f(A)$ is M-normal.*

Proof. Let p be any polynomial that evaluates f at A. Then

$$f(A)^\star = p(A)^\star = \begin{cases} p(A^\star) & \text{for bilinear forms,} \\ \overline{p}(A^\star) & \text{for sesquilinear forms.} \end{cases}$$

Continuing with the two cases,

$$\begin{aligned} f(A)f(A)^\star &= \begin{cases} p(A)p(A^\star) = p(A^\star)p(A) \\ p(A)\overline{p}(A^\star) = \overline{p}(A^\star)p(A) \end{cases} \quad (\text{since } AA^\star = A^\star A) \\ &= f(A)^\star f(A). \quad ∎ \end{aligned}$$

Theorem 3.14 generalizes a result of Gohberg, Lancaster, and Rodman [8, Thm. I.6.3], which states that if $A \in \mathbb{G}$ or $A = A^\star$ with respect to a Hermitian sesquilinear form then $f(A)$ is M-normal.

4. Connections between the matrix sign function, the generalized polar decomposition, and the matrix square root. Having identified the matrix sign function and square root as structure preserving for all groups, we now consider computational matters. We show in this section that the matrix sign function and square

root are intimately connected with each other and also with the generalized polar decomposition. Given a scalar product on $\mathbb{K}^n$ with adjoint $(\cdot)^\star$, a generalized polar decomposition of a matrix $A \in \mathbb{K}^{n\times n}$ is a decomposition $A = WS$, where W is an automorphism and S is self-adjoint with spectrum contained in the open right half-plane; that is, $W^\star = W^{-1}$, $S^\star = S$, and $\operatorname{sign}(S) = I$. The existence and uniqueness of a generalized polar decomposition are described in the next result, which extends [15, Thm. 4.1].

THEOREM 4.1 (generalized polar decomposition). *With respect to an arbitrary scalar product on $\mathbb{K}^n$, a matrix $A \in \mathbb{K}^{n\times n}$ has a generalized polar decomposition $A = WS$ if and only if $(A^\star)^\star = A$ and $A^\star A$ has no eigenvalues on $\mathbb{R}^-$. When such a factorization exists, it is unique.*

Proof. ($\Rightarrow$) Note first that if the factorization exists then

$$(A^\star)^\star = (S^\star W^\star)^\star = (SW^{-1})^\star = W^{-\star}S^\star = WS = A.$$

Also we must have

$$A^\star A = S^\star W^\star W S = S^\star S = S^2. \tag{4.1}$$

But if $\operatorname{sign}(S) = I$ is to hold then the only possible choice for S is $S = (A^\star A)^{1/2}$, and this square root exists only if $A^\star A$ has no eigenvalues on $\mathbb{R}^-$.

($\Leftarrow$) Letting $S = (A^\star A)^{1/2}$, the condition $\operatorname{sign}(S) = I$ is automatically satisfied, but we need also to show that $S^\star = S$. First, note that for any B with no eigenvalues on $\mathbb{R}^-$ we have $(B^\star)^{1/2} = (B^{1/2})^\star$. Indeed $(B^{1/2})^\star$ is a square root of $B^\star$, because $(B^{1/2})^\star(B^{1/2})^\star = (B^{1/2}\cdot B^{1/2})^\star = B^\star$, and the fact that $(B^{1/2})^\star$ is similar to $(B^{1/2})^T$ (for bilinear forms) or $(B^{1/2})^*$ (for sesquilinear forms) implies that $(B^{1/2})^\star$ must be the principal square root. Then, using the assumption that $(A^\star)^\star = A$, we have

$$S^\star = \left((A^\star A)^{1/2}\right)^\star = \left((A^\star A)^\star\right)^{1/2} = (A^\star A)^{1/2} = S.$$

Finally, the uniquely defined matrix $W = AS^{-1}$ satisfies

$$W^\star W = (AS^{-1})^\star(AS^{-1}) = S^{-\star}(A^\star A)S^{-1} = S^{-1}(S^2)S^{-1} = I,$$

using (4.1), and so $W \in \mathbb{G}$. ∎

For many scalar products, including all those in Table 2.1, $(A^\star)^\star = A$ holds *for all* $A \in \mathbb{K}^{n\times n}$, in which case we say that the adjoint is involutory. It can be shown that the adjoint is involutory if and only if $M^T = \pm M$ for bilinear forms and $M^* = \alpha M$ with $|\alpha| = 1$ for sesquilinear forms [26]. But even for scalar products for which the adjoint is not involutory, there are always many matrices A for which $(A^\star)^\star = A$, as the next result shows. We omit the straightforward proof.

LEMMA 4.2. *Let $\mathbb{G}$ be the automorphism group of a scalar product. The condition*

$$(A^\star)^\star = A \tag{4.2}$$

is satisfied if $A \in \mathbb{G}$, $A = A^\star$, or $A = -A^\star$. Moreover, arbitrary products and linear combinations of matrices satisfying (4.2) *also satisfy* (4.2).

The generalized polar decomposition as we have defined it is closely related to the polar decompositions corresponding to Hermitian sesquilinear forms on $\mathbb{C}^n$ studied by Bolshakov et al. [2], [3], the symplectic polar decomposition introduced by Ikramov [17], and the polar decompositions corresponding to symmetric bilinear forms on $\mathbb{C}^n$ considered by Kaplansky [19]. In these papers the self-adjoint factor S may or may

not be required to satisfy additional conditions, but $\operatorname{sign}(S) = I$ is not one of those considered. The connections established below between the matrix sign function, the principal matrix square root, and the generalized polar decomposition as we have defined it, suggest that $\operatorname{sign}(S) = I$ is the appropriate extra condition for a generalized polar decomposition of computational use.

The following result, which we have not found in the literature, is the basis for the connections to be established.

LEMMA 4.3. *Let $A, B \in \mathbb{C}^{n\times n}$ and suppose that AB (and hence also BA) has no eigenvalues on $\mathbb{R}^-$. Then*

$$\operatorname{sign}\left(\begin{bmatrix} 0 & A \\ B & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & C \\ C^{-1} & 0 \end{bmatrix},$$

where $C = A(BA)^{-1/2}$.

Proof. The matrix $P = \left[\begin{smallmatrix} 0 & A \\ B & 0 \end{smallmatrix}\right]$ cannot have any eigenvalues on the imaginary axis, because if it did, then $P^2 = \left[\begin{smallmatrix} AB & 0 \\ 0 & BA \end{smallmatrix}\right]$ would have an eigenvalue on $\mathbb{R}^-$. Hence $\operatorname{sign}(P)$ is defined and

$$\begin{aligned} \operatorname{sign}(P) = P(P^2)^{-1/2} &= \begin{bmatrix} 0 & A \\ B & 0 \end{bmatrix} \begin{bmatrix} AB & 0 \\ 0 & BA \end{bmatrix}^{-1/2} \\ &= \begin{bmatrix} 0 & A \\ B & 0 \end{bmatrix} \begin{bmatrix} (AB)^{-1/2} & 0 \\ 0 & (BA)^{-1/2} \end{bmatrix} \\ &= \begin{bmatrix} 0 & A(BA)^{-1/2} \\ B(AB)^{-1/2} & 0 \end{bmatrix} =: \begin{bmatrix} 0 & C \\ D & 0 \end{bmatrix}. \end{aligned}$$

Since the square of the matrix sign of any matrix is the identity,

$$I = (\operatorname{sign}(P))^2 = \begin{bmatrix} 0 & C \\ D & 0 \end{bmatrix}^2 = \begin{bmatrix} CD & 0 \\ 0 & DC \end{bmatrix};$$

thus $D = C^{-1}$. Alternatively, Corollary 2.2 may be used to see more directly that $CD = A(BA)^{-1/2}B(AB)^{-1/2}$ is equal to I. ∎

Two important special cases of Lemma 4.3 are, for $A \in \mathbb{C}^{n\times n}$ with no eigenvalues on $\mathbb{R}^-$ [13],

$$\operatorname{sign}\left(\begin{bmatrix} 0 & A \\ I & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & A^{1/2} \\ A^{-1/2} & 0 \end{bmatrix}, \tag{4.3}$$

and, for nonsingular $A \in \mathbb{C}^{n\times n}$ [12],

$$\operatorname{sign}\left(\begin{bmatrix} 0 & A \\ A^* & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & U \\ U^* & 0 \end{bmatrix}, \tag{4.4}$$

where $A = UH$ is the polar decomposition. A further special case, which generalizes (4.4), is given in the next result.

COROLLARY 4.4. *If $A \in \mathbb{K}^{n\times n}$ has a generalized polar decomposition $A = WS$ then*

$$\operatorname{sign}\left(\begin{bmatrix} 0 & A \\ A^\star & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & W \\ W^\star & 0 \end{bmatrix}. \tag{4.5}$$

Proof. Lemma 4.3 gives

$$\operatorname{sign}\left(\begin{bmatrix} 0 & A \\ A^\star & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & C \\ C^{-1} & 0 \end{bmatrix},$$

where $C = A(A^\star A)^{-1/2}$. Using the given generalized polar decomposition, $C = WS \cdot S^{-1} = W$ and therefore

$$\operatorname{sign}\left(\begin{bmatrix} 0 & A \\ A^\star & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & W \\ W^{-1} & 0 \end{bmatrix} = \begin{bmatrix} 0 & W \\ W^\star & 0 \end{bmatrix}. \qquad \square$$

The significance of (4.3)–(4.5) is that they enable results and iterations for the sign function to be translated into results and iterations for the square root and generalized polar decomposition. For example, Roberts' integral formula [28], $\operatorname{sign}(A) = (2/\pi)A\int_0^\infty (t^2 I + A^2)^{-1}\,dt$, translates, via (4.5), into an integral representation for the generalized polar factor W:

$$W = \frac{2}{\pi} A \int_0^\infty (t^2 I + A^\star A)^{-1}\,dt.$$

Our interest in the rest of this section is in deriving iterations, beginning with a family of iterations for the matrix square root.[3]

THEOREM 4.5. *Suppose the matrix A has no eigenvalues on $\mathbb{R}^-$, so that $A^{1/2}$ exists. Let g be any matrix function of the form $g(X) = Xh(X^2)$ such that the iteration $X_{k+1} = g(X_k)$ converges to* $\operatorname{sign}(X_0)$ *with order of convergence m whenever* $\operatorname{sign}(X_0)$ *is defined. Then in the coupled iteration*

$$\begin{aligned} Y_{k+1} &= Y_k h(Z_k Y_k), \qquad Y_0 = A, \\ Z_{k+1} &= h(Z_k Y_k) Z_k, \qquad Z_0 = I, \end{aligned} \tag{4.6}$$

$Y_k \to A^{1/2}$ and $Z_k \to A^{-1/2}$ as $k \to \infty$, both with order of convergence m, Y_k commutes with Z_k, and $Y_k = AZ_k$ for all k. Moreover, if g is structure preserving for an automorphism group $\mathbb{G}$, then iteration (4.6) *is also structure preserving for $\mathbb{G}$, that is, $A \in \mathbb{G}$ implies $Y_k, Z_k \in \mathbb{G}$ for all k.*

Proof. Observe that

$$\begin{aligned} g\left(\begin{bmatrix} 0 & Y_k \\ Z_k & 0 \end{bmatrix}\right) &= \begin{bmatrix} 0 & Y_k \\ Z_k & 0 \end{bmatrix} h\left(\begin{bmatrix} Y_k Z_k & 0 \\ 0 & Z_k Y_k \end{bmatrix}\right) \\ &= \begin{bmatrix} 0 & Y_k \\ Z_k & 0 \end{bmatrix} \begin{bmatrix} h(Y_k Z_k) & 0 \\ 0 & h(Z_k Y_k) \end{bmatrix} \\ &= \begin{bmatrix} 0 & Y_k h(Z_k Y_k) \\ Z_k h(Y_k Z_k) & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & Y_k h(Z_k Y_k) \\ h(Z_k Y_k) Z_k & 0 \end{bmatrix} = \begin{bmatrix} 0 & Y_{k+1} \\ Z_{k+1} & 0 \end{bmatrix}, \end{aligned}$$

where the penultimate equality follows from Corollary 2.2. The initial conditions $Y_0 = A$ and $Z_0 = I$ together with (4.3) now imply that Y_k and Z_k converge to $A^{1/2}$ and $A^{-1/2}$, respectively. It is easy to see that Y_k and Z_k are polynomials in A for all k,

[3] We note that if we generalize to $Z_0 = B$ in (4.6), where BA has no eigenvalues on $\mathbb{R}^-$, then $Y_k \to A(BA)^{-1/2}$, which is a solution of the special Riccati equation $XBX = A$, while $Z_k \to B(AB)^{-1/2}$, which solves $XAX = B$.

and hence Y_k commutes with Z_k. Then $Y_k = AZ_k$ follows by induction. The order of convergence of the coupled iteration (4.6) is clearly the same as that of the sign iteration from which it arises.

Finally, if g is structure preserving for $\mathbb{G}$ and $A \in \mathbb{G}$, then we can show inductively that $Y_k, Z_k \in \mathbb{G}$ for all k. Clearly $Y_0, Z_0 \in \mathbb{G}$. Assuming that $Y_k, Z_k \in \mathbb{G}$, then $Z_k Y_k \in \mathbb{G}$. Since $\operatorname{sign}\left[\begin{smallmatrix} 0 & Y_k \\ Z_k & 0 \end{smallmatrix}\right] = \operatorname{sign}\left[\begin{smallmatrix} 0 & A \\ I & 0 \end{smallmatrix}\right]$, we know that $P = \left[\begin{smallmatrix} 0 & Y_k \\ Z_k & 0 \end{smallmatrix}\right]$ has no imaginary eigenvalues and hence that $P^2 = \left[\begin{smallmatrix} Y_k Z_k & 0 \\ 0 & Z_k Y_k \end{smallmatrix}\right]$ has no eigenvalues on $\mathbb{R}^-$. Thus $(Z_k Y_k)^{1/2}$ exists, and from section 3 we know that $(Z_k Y_k)^{1/2} \in \mathbb{G}$. But for any $X \in \mathbb{G}$, $g(X) \in \mathbb{G}$ and hence $h(X^2) = X^{-1} g(X) \in \mathbb{G}$. Thus with $X = (Z_k Y_k)^{1/2}$, we see that $h(X^2) = h(Z_k Y_k) \in \mathbb{G}$, and therefore $Y_{k+1}, Z_{k+1} \in \mathbb{G}$. ∎

The connection between sign iterations and square root iterations has been used previously [13], but only for some particular g. By contrast, Theorem 4.5 is very general, since all commonly used sign iteration functions have the form $g(X) = X h(X^2)$ considered here. Note that the commutativity of Y_k and Z_k allows several variations of (4.6); the one we have chosen has the advantage that it requires only one evaluation of h per iteration. We have deliberately avoided using commutativity properties in deriving the iteration within the proof above (instead, we invoked Corollary 2.2). In particular, we did not rewrite the second part of the iteration in the form $Z_{k+1} = Z_k h(Z_k Y_k)$, which is arguably more symmetric with the first part. The reason is that experience suggests that exploiting commutativity when deriving matrix iterations can lead to numerical instability (see, e.g., [11]). Indeed we will show in section 5 that while (4.6) is numerically stable, the variant just mentioned is not.

We now exploit the connection in Corollary 4.4 between the sign function and the generalized polar decomposition. The corollary suggests that we apply iterations for the matrix sign function to

$$X_0 = \begin{bmatrix} 0 & A \\ A^\star & 0 \end{bmatrix},$$

so just as in Theorem 4.5 we consider iteration functions of the form $g(X) = X h(X^2)$. It is possible, though nontrivial, to prove by induction that all the iterates X_k of such a g have the form

$$X_k = \begin{bmatrix} 0 & Y_k \\ Y_k^\star & 0 \end{bmatrix} \tag{4.7}$$

with $(Y_k^\star)^\star = Y_k$, and that $Y_{k+1} = Y_k h(Y_k^\star Y_k)$—under an extra assumption on g in the sesquilinear case. Corollary 4.4 then implies that Y_k converges to the generalized polar factor W of A. While this approach is a useful way to derive the iteration for W, a shorter and more direct demonstration of the claimed properties is possible, as we now show.

THEOREM 4.6. *Suppose the matrix A has a generalized polar decomposition $A = WS$ with respect to a given scalar product. Let g be any matrix function of the form $g(X) = X h(X^2)$ such that the iteration $X_{k+1} = g(X_k)$ converges to* $\operatorname{sign}(X_0)$ *with order of convergence m whenever* $\operatorname{sign}(X_0)$ *is defined. For sesquilinear forms assume that g also satisfies* (d) *in Theorem* 3.1 *for all matrices in its domain. Then the iteration*

$$Y_{k+1} = Y_k h(Y_k^\star Y_k), \qquad Y_0 = A \tag{4.8}$$

converges to W with order of convergence m.

Proof. Let $X_{k+1} = g(X_k)$ with $X_0 = S$, so that $\lim_{k\to\infty} X_k = \operatorname{sign}(S) = I$. We claim that $X_k^\star = X_k$ and $Y_k = WX_k$ for all k. These equalities are trivially true for $k = 0$. Assuming that they are true for k, we have

$$X_{k+1}^\star = g(X_k)^\star = g(X_k^\star) = g(X_k) = X_{k+1}$$

and

$$Y_{k+1} = WX_k h(X_k^\star W^\star W X_k) = WX_k h(X_k^2) = WX_{k+1}.$$

The claim follows by induction. Hence $\lim_{k\to\infty} Y_k = W \lim_{k\to\infty} X_k = W$. The order of convergence is readily seen to be m. ∎

Theorem 4.6 shows that iterations for the matrix sign function automatically yield iterations for the generalized polar factor W. The next result reveals that the square root of a matrix in an automorphism group is the generalized polar factor W of a related matrix. Consequently, iterations for W also lead to iterations for the matrix square root, although only for matrices in automorphism groups. We will take up this topic again in section 6.

THEOREM 4.7. *Let $\mathbb{G}$ be the automorphism group of a scalar product and $A \in \mathbb{G}$. If A has no eigenvalues on $\mathbb{R}^-$, then $I+A = WS$ with $W = A^{1/2}$ and $S = A^{-1/2}+A^{1/2}$ is the generalized polar decomposition of $I+A$ with respect to the given scalar product.*

Proof. Clearly, $I + A = WS$ and $W^\star = A^{\star/2} = A^{-1/2} = W^{-1}$. It remains to show that $S^\star = S$ and $\operatorname{sign}(S) = I$. We have

$$S^\star = A^{-\star/2} + A^{\star/2} = A^{1/2} + A^{-1/2} = S.$$

Moreover, the eigenvalues of S are of the form $\mu = \lambda^{-1} + \lambda$, where $\lambda \in \Lambda(A^{1/2})$ is in the open right half-plane. Clearly μ is also in the open right half-plane, and hence $\operatorname{sign}(S) = I$. ∎

Note that Theorem 4.7 does not make any assumption on the scalar product or its associated adjoint. The condition $(B^\star)^\star = B$ that is required to apply Theorem 4.1 is automatically satisfied for $B = I + A$, since $A \in \mathbb{G}$ implies that $I + A$ is one of the matrices in Lemma 4.2.

Theorem 4.7 appears in Cardoso, Kenney, and Silva Leite [5, Thm. 6.3] for real bilinear forms only and with the additional assumption that the matrix M of the scalar product is symmetric positive definite.

5. Stability analysis of coupled square root iterations. Before investigating any specific iterations from among the families obtained in the previous section, we carry out a stability analysis of the general iteration (4.6) of Theorem 4.5. A whole section is devoted to this analysis for two reasons. First, as is well known, minor rewriting of matrix iterations can completely change their stability properties [11], [13]. As already noted, (4.6) can be rewritten in various ways using commutativity and/or Corollary 2.2, and it is important to know that a choice of form motivated by computational cost considerations does not sacrifice stability. Second, we are able to give a stability analysis of (4.6) in its full generality, and in doing so introduce a technique that is novel in this context and should be of wider use in analyzing the stability of matrix iterations.

We begin by slightly changing the notation of Theorem 4.5. Consider matrix functions of the form $g(X) = Xh(X^2)$ that compute the matrix sign by iteration and the related function

$$G(Y, Z) = \begin{bmatrix} g_1(Y, Z) \\ g_2(Y, Z) \end{bmatrix} = \begin{bmatrix} Yh(ZY) \\ h(ZY)Z \end{bmatrix}. \tag{5.1}$$

Iterating G starting with $(Y, Z) = (A, I)$ produces the coupled iteration (4.6), which we know converges to $(A^{1/2}, A^{-1/2})$. Recall that the Fréchet derivative of a map $F : \mathbb{C}^{m\times n} \to \mathbb{C}^{m\times n}$ at a point $X \in \mathbb{C}^{m\times n}$ is a linear mapping $L_X : \mathbb{C}^{m\times n} \to \mathbb{C}^{m\times n}$ such that for all $E \in \mathbb{C}^{m\times n}$ [6], [29],

$$F(X + E) - F(X) - L_X(E) = o(\|E\|).$$

For our purposes it will not matter whether L_X is $\mathbb{C}$-linear or only $\mathbb{R}$-linear.

Our aim is to find the Fréchet derivative of the map G at the point $(Y, Z) = (A^{1/2}, A^{-1/2})$, or, more generally, at any point of the form (B, B^{-1}); later, these points will all be seen to be fixed points of the map G. We denote the Fréchet derivative of G by dG, the derivative at a particular point (A, B) by $dG_{(A,B)}$, and the matrix inputs to dG by dY and dZ. With this notation, we have

$$dG_{(Y,Z)}(dY, dZ) = \begin{bmatrix} dg_1(dY, dZ) \\ dg_2(dY, dZ) \end{bmatrix} = \begin{bmatrix} Y dh_{ZY}(ZdY + dZ \cdot Y) + dY \cdot h(ZY) \\ dh_{ZY}(ZdY + dZ \cdot Y) \cdot Z + h(ZY)dZ \end{bmatrix}.$$

At the point $(Y, Z) = (B, B^{-1})$ this simplifies to

$$dG_{(B,B^{-1})}(dY, dZ) = \begin{bmatrix} B\, dh_I(B^{-1}dY + dZ \cdot B) + dY \cdot h(I) \\ dh_I(B^{-1}dY + dZ \cdot B) \cdot B^{-1} + h(I)dZ \end{bmatrix}. \tag{5.2}$$

In order to further simplify this expression we need to know more about $h(I)$ and dh_I. We give a preliminary lemma and then exploit the fact that h is part of a function that computes the matrix sign.

LEMMA 5.1. *For any matrix function $F(X)$ with underlying scalar function f that is analytic at $z = 1$, the Fréchet derivative of F at the matrix I is just scalar multiplication by $f'(1)$, that is, $dF_I(E) = f'(1)E$.*

Proof. Expand the scalar function f as a convergent power series about $z = 1$: $f(z) = \sum_{k=0}^{\infty} b_k(z - 1)^k$, where $b_k = f^{(k)}(1)/k!$. Then

$$F(I + E) - F(I) = \sum_{k=0}^{\infty} b_k E^k - b_0 I = b_1 E + O(\|E\|^2).$$

Thus $dF_I(E) = b_1 E = f'(1)E$. ∎

LEMMA 5.2. *Suppose h is part of a matrix function of the form $g(X) = X h(X^2)$ such that the iteration $X_{k+1} = g(X_k)$ converges superlinearly to $\operatorname{sign}(X_0)$ whenever $\operatorname{sign}(X_0)$ exists. If the scalar function h is analytic at $z = 1$, then $h(I) = I$ and $dh_I(E) = -\frac{1}{2}E$.*

Proof. Since $\operatorname{sign}(I) = I$, I is a fixed point of the iteration, so $g(I) = I$ and hence $h(I) = I$.

At the scalar level, $g(x) = xh(x^2)$ and $g'(x) = 2x^2h'(x^2) + h(x^2)$, so $g'(1) = 2h'(1) + h(1)$. But $h(I) = I$ implies $h(1) = 1$, so $g'(1) = 2h'(1) + 1$. Now we are assuming that the iterates of g converge superlinearly to $\operatorname{sign}(X_0)$, so in particular we know that a neighborhood of 1 contracts superlinearly to 1 under iteration by g. From fixed point iteration theory this means that $g'(1) = 0$. Hence $h'(1) = -\frac{1}{2}$ and, using Lemma 5.1, $dh_I(E) = h'(1)E = -\frac{1}{2}E$. ∎

Because $h(I) = I$, it is now clear that any point (B, B^{-1}) is a fixed point for G. Furthermore, our knowledge of $h(I)$ and dh_I allows us to complete the simplification of dG, continuing from (5.2):

$$dG_{(B,B^{-1})}(dY,dZ) = \begin{bmatrix} -\frac{1}{2}dY - \frac{1}{2}B\,dZ\,B + dY \\ -\frac{1}{2}B^{-1}dY\,B^{-1} - \frac{1}{2}dZ + dZ \end{bmatrix} = \begin{bmatrix} \frac{1}{2}dY - \frac{1}{2}B\,dZ\,B \\ \frac{1}{2}dZ - \frac{1}{2}B^{-1}dY\,B^{-1} \end{bmatrix}.$$

A straightforward computation shows that $dG_{(B,B^{-1})}$ is idempotent and hence is a projection. We summarize our findings in a theorem.

THEOREM 5.3. *Consider any iteration of the form* (4.6) *and its associated mapping*

$$G(Y,Z) = \begin{bmatrix} Yh(ZY) \\ h(ZY)Z \end{bmatrix},$$

where $X_{k+1} = g(X_k) = X_k h(X_k^2)$ *is any superlinearly convergent iteration for the matrix sign such that the scalar function* h *is analytic at* $z = 1$. *Then any matrix pair of the form* $P = (B, B^{-1})$ *is a fixed point for* G, *and the Fréchet derivative of* G *at* P *is given by*

$$dG_P(E,F) = \frac{1}{2}\begin{bmatrix} E - BFB \\ F - B^{-1}EB^{-1} \end{bmatrix}.$$

The derivative map dG_P *is idempotent, that is,* $dG_P \circ dG_P = dG_P$.

Following Cheng et al. [7], we define an iteration $X_{k+1} = g(X_k)$ to be stable in a neighborhood of a fixed point $X = g(X)$ if for $X_0 := X + H_0$, with arbitrary H_0, the errors $H_k := X_k - X$ satisfy

$$H_{k+1} = L_X(H_k) + O(\|H_k\|^2), \tag{5.3}$$

where L_X is a linear operator (necessarily the Fréchet derivative of g at X) with bounded powers, that is, there exists a constant c such that for all $s > 0$ and arbitrary H of unit norm, $\|L_X^s(H)\| \le c$. Note that the iterations we are considering have a specified X_0 and so the convergence analysis in section 4 says nothing about the effect of arbitrary errors H_k in the X_k. In practice, such errors are of course introduced by the effects of roundoff. The significance of Theorem 5.3 is that it shows that any iteration belonging to the broad class (4.6) is stable, for L_X is here idempotent and hence trivially has bounded powers.

A further use of our analysis is to predict the limiting accuracy of the iteration in floating point arithmetic, that is, the smallest error we can expect. Consider $X_0 = X + H_0$ with $\|H_0\| \le u\|X\|$, where u is the unit roundoff, so that X_0 can be thought of as X rounded to floating point arithmetic. Then from (5.3) we have $\|H_1\| \lesssim \|L_X(H_0)\|$, and so an estimate of the absolute limiting accuracy is any bound for $\|L_X(H_0)\|$. In the case of iteration (4.6), a suitable bound is, from Theorem 5.3 with $B = A^{1/2}$,

$$\max\left\{\|E_0\| + \|A^{1/2}\|^2\|F_0\|, \|F_0\| + \|A^{-1/2}\|^2\|E_0\|\right\},$$

where $\|E_0\| \le \|A^{1/2}\|u$ and $\|F_0\| \le \|A^{-1/2}\|u$. For any of the classical groups in Table 2.1, M is unitary and so $A \in \mathbb{G}$ implies $\|A^{1/2}\|_2 = \|A^{-1/2}\|_2$, by (2.2) (since

$A^{1/2} \in \mathbb{G}$). Hence this bound is just $\|A^{1/2}\|_2(1 + \|A^{1/2}\|_2^2)u$, giving an estimate for the *relative* limiting accuracy of $(1 + \|A^{1/2}\|_2^2)u$.

The Fréchet derivative-based analysis of this section would be even more useful if it also allowed us to identify otherwise plausible iterations that are unstable. To see that it does, consider the mathematically equivalent variant of (4.6),

$$\begin{aligned} Y_{k+1} &= Y_k h(Z_k Y_k), \qquad Y_0 = A, \\ Z_{k+1} &= Z_k h(Z_k Y_k), \qquad Z_0 = I, \end{aligned} \tag{5.4}$$

mentioned earlier as being arguably more symmetric but of questionable stability since its derivation relies on commutativity properties. For this iteration we define the map $\widetilde{G}(Y, Z) = \left[\begin{smallmatrix} Yh(ZY) \\ Zh(ZY) \end{smallmatrix}\right]$, analogous to the map G for iteration (4.6), and see by a calculation similar to the one above that

$$d\widetilde{G}_P(E, F) = \frac{1}{2}\begin{bmatrix} E - BFB \\ 2F - B^{-1}FB - B^{-2}E \end{bmatrix}. \tag{5.5}$$

The following lemma, whose proof we omit, shows that for many B the map $d\widetilde{G}_P$ has an eigenvalue of modulus exceeding 1 and hence does not have bounded powers; the iteration is then unstable according to our definition.

LEMMA 5.4. *If α and β are any two eigenvalues of B then $\gamma = \frac{1}{2}(1 - \frac{\alpha}{\beta})$ is an eigenvalue for $d\widetilde{G}_P$ in* (5.5)*, where $P = (B, B^{-1})$.*

The stability and instability, respectively, of particular instances of iterations (4.6) and (5.4) are confirmed in the numerical experiments of section 7.

Finally, we note that the following analogue of Theorem 5.3 can be proved for the iterations computing the generalized polar factor W described in Theorem 4.6.

THEOREM 5.5. *Consider any iteration of the form* (4.8) *and the associated mapping $f(Y) = Yh(Y^\star Y)$, where $X_{k+1} = g(X_k) = X_k h(X_k^2)$ is any superlinearly convergent iteration for the matrix sign such that the scalar function h is analytic at $z = 1$. Then any $B \in \mathbb{G}$ is a fixed point for f, and the Fréchet derivative of f at B is given by $df_B(E) = \frac{1}{2}(E - BE^\star B)$. If the underlying scalar product has an involutory adjoint, then the derivative map df_B is idempotent.*

As an immediate consequence we see that any iteration of the form (4.8) is stable,[4] at least when the adjoint is involutory (see the remarks preceding Lemma 4.2 for details of when this condition holds). Special cases of this include the unitary polar factor iterations developed in [15] and the iteration (6.7) for the square root of matrices in $\mathbb{G}$ derived in the next section.

6. Iterations for the matrix square root. We now use the theory developed above to derive some specific new iterations for computing the square root of a matrix in an automorphism group. We assume throughout that A has no eigenvalues on $\mathbb{R}^-$, so that $A^{1/2}$ is defined. First, we recall the well-known Newton iteration

$$X_{k+1} = \frac{1}{2}(X_k + X_k^{-1}A), \qquad X_0 = A, \tag{6.1}$$

which can be thought of as a generalization to matrices of Heron's iteration for the square root of a scalar. This iteration converges quadratically to $A^{1/2}$, but it is numerically unstable and therefore not of practical use [11], [23]. There has consequently been much interest in deriving numerically stable alternatives.

[4]Because of the presence of the adjoint in iteration (4.8), the map L_X in (5.3) is no longer complex linear in the sesquilinear case, but it is a real linear map and hence we can still deduce stability.

We first derive a structure-preserving iteration. We apply Theorem 4.5 to the family of structure-preserving matrix sign function iterations identified by Higham et al. [15], which comprises the main diagonal of a table of Padé-based iterations discovered by Kenney and Laub [20].

THEOREM 6.1. *Let $A \in \mathbb{K}^{n\times n}$ and consider the iterations*

$$Y_{k+1} = Y_k\, p_m(I - Z_kY_k)\,[\mathrm{rev}p_m(I - Z_kY_k)]^{-1}, \qquad Y_0 = A, \tag{6.2a}$$

$$Z_{k+1} = p_m(I - Z_kY_k)\,[\mathrm{rev}p_m(I - Z_kY_k)]^{-1}Z_k, \qquad Z_0 = I, \tag{6.2b}$$

where $p_m(t)$ is the numerator in the $[m/m]$ Padé approximant to $(1-t)^{-1/2}$ and $m \geq 1$. Assume that A has no eigenvalues on $\mathbb{R}^-$ and $A \in \mathbb{G}$, where $\mathbb{G}$ is any automorphism group. Then $Y_k \in \mathbb{G}$, $Z_k \in \mathbb{G}$, and $Y_k = AZ_k$ for all k, and $Y_k \to A^{1/2}$, $Z_k \to A^{-1/2}$, both with order of convergence $2m + 1$.

Proof. It was shown in [15] that the iteration $X_{k+1} = X_k\, p_m(I - X_k^2)\,[\mathrm{rev}p_m(I - X_k^2)]^{-1}$, with $X_0 = A$, is on the main diagonal of the Padé table in [20] and so converges to $\mathrm{sign}(A)$ with order of convergence $2m + 1$. This iteration was shown in [15] to be structure preserving, a property that can also be seen from Theorem 3.13(e). The theorem therefore follows immediately from Theorem 4.5. ∎

The polynomial $p_m(1 - x^2)$ in Theorem 6.1 can be obtained by taking the odd part of $(1 + x)^{2m+1}$ and dividing through by x [20]. The first two polynomials are $p_1(1-x^2) = x^2+3$ and $p_2(1-x^2) = x^4+10x^2+5$. The cubically converging iteration $(m = 1)$ is therefore

$$Y_{k+1} = Y_k(3I + Z_kY_k)(I + 3Z_kY_k)^{-1}, \qquad Y_0 = A, \tag{6.3a}$$

$$Z_{k+1} = (3I + Z_kY_k)(I + 3Z_kY_k)^{-1}Z_k, \qquad Z_0 = I. \tag{6.3b}$$

A rearrangement of these formulae that can be evaluated in fewer flops is the continued fraction form, adapted from [15],

$$Y_{k+1} = \frac{1}{3}Y_k\big[I + 8(I + 3Z_kY_k)^{-1}\big], \qquad Y_0 = A, \tag{6.4a}$$

$$Z_{k+1} = \frac{1}{3}\big[I + 8(I + 3Z_kY_k)^{-1}\big]Z_k, \qquad Z_0 = I. \tag{6.4b}$$

This iteration can be implemented in two ways: using three matrix multiplications and one (explicit) matrix inversion per iteration, or with one matrix multiplication and two solutions of matrix equations involving coefficient matrices that are transposes of each other. The latter approach has the smaller operation count, but the former could be faster in practice as it is richer in matrix multiplication, which is a particularly efficient operation on modern computers.

A related family[5] of coupled iterations for the square root was derived by Higham [13] from the first superdiagonal of Kenney and Laub's Padé table. However, unlike (6.2), that family is not structure preserving: when $A \in \mathbb{G}$ the iterates do not stay in the group.

With the aid of Theorem 4.7 we can derive iterations that, while not structure preserving, are specifically designed for matrices in automorphism groups. Theorem 4.7 says that computing the square root of $A \in \mathbb{G}$ is equivalent to computing the

[5] In [13], iteration (2.8) therein was rewritten using commutativity to obtain a more efficient form (2.10), which was found to be unstable. This form is (essentially) a particular case of (5.4). If instead (2.8) is rewritten using Corollary 2.2, as we did in deriving (4.6) in section 4, efficiency is gained without the loss of stability.

generalized polar factor W of $I + A$. Theorem 4.6 says that any of a wide class of iterations for the sign of a matrix yields a corresponding iteration for the generalized polar factor W of the matrix. The simplest application of this result is to the Newton iteration for the sign function,

$$X_{k+1} = \frac{1}{2}\big(X_k + X_k^{-1}\big), \qquad X_0 = A. \tag{6.5}$$

Applying Theorem 4.6 we deduce that for any A having a generalized polar decomposition $A = WS$, the iteration

$$Y_{k+1} = \frac{1}{2}\big(Y_k + Y_k^{-\star}\big), \qquad Y_0 = A \tag{6.6}$$

is well-defined and Y_k converges quadratically to W. This iteration is also analyzed by Cardoso, Kenney, and Silva Leite [5, sect. 4], who treat real bilinear forms only and assume that the matrix M underlying the bilinear form is orthogonal and either symmetric or skew-symmetric. Higham [14] analyzes (6.6) in the special case of the pseudo-orthogonal group. In the special case of the real orthogonals, $M = I$, and (6.6) reduces to the well-known Newton iteration for the orthogonal polar factor [10].

On invoking Theorem 4.7 we obtain the matrix square root iteration in the next result.

THEOREM 6.2. *Let $\mathbb{G}$ be any automorphism group and $A \in \mathbb{G}$. If A has no eigenvalues on $\mathbb{R}^-$ then the iteration*

$$\begin{aligned} Y_{k+1} &= \frac{1}{2}\big(Y_k + Y_k^{-\star}\big) \\ &= \begin{cases} \frac{1}{2}\big(Y_k + M^{-1}Y_k^{-T}M\big) & \textit{for bilinear forms,} \\ \frac{1}{2}\big(Y_k + M^{-1}Y_k^{-*}M\big) & \textit{for sesquilinear forms,} \end{cases} \end{aligned} \tag{6.7}$$

with starting matrix $Y_1 = \frac{1}{2}(I + A)$, is well defined and Y_k converges quadratically to $A^{1/2}$. The iterates Y_k are identical to the X_k $(k \geq 1)$ in (6.1) generated by Newton's method.

Proof. Only the last part remains to be explained. It is easy to show by induction that $X_k^\star = A^{-1}X_k$ $(k \geq 1)$, from which $X_k = Y_k$ $(k \geq 1)$ follows by a second induction. □

Note that the factor $\frac{1}{2}$ in Y_1 is chosen to ensure that $Y_k \equiv X_k$ for $k \geq 1$; since $\frac{1}{2}(I + A) = W(\frac{1}{2}S)$, W is unaffected by this factor.

Theorem 6.2 shows that for A in an automorphism group the Newton iteration (6.1) can be rewritten in an alternative form—one that has much better numerical stability properties, as we will show below.

The iteration in Theorem 6.2 is also investigated by Cardoso, Kenney, and Silva Leite [5, sect. 6], with the same assumptions on $\mathbb{G}$ as mentioned above for their treatment of (6.6).

If M is a general matrix then the operation count for (6.7) is higher than that for the Newton iteration (6.1). However, for all the classical groups M is a permutation of diag(± 1) (see Table 2.1) and multiplication by M^{-1} and M is therefore of trivial cost; for these groups the cost of iteration (6.7) is one matrix inversion per iteration, which operation counts show is about 75% of the cost per iteration of (6.1) and 30% of that for (6.4).

Matrix Newton iterations benefit from scaling when the starting matrix A is far from the limit. Much is known about scalings for the sign function iteration (6.5) of the form

$$X_{k+1} = \frac{1}{2}\bigl(\alpha_k X_k + \alpha_k^{-1} X_k^{-1}\bigr), \qquad X_0 = A; \tag{6.8}$$

see Kenney and Laub [21]. The corresponding scaled version of (6.7) is

$$Y_{k+1} = \frac{1}{2}\bigl(\gamma_k Y_k + (\gamma_k Y_k)^{-\star}\bigr), \qquad Y_1 = \frac{1}{2}(I + A). \tag{6.9}$$

By considering the discussion just before the proof of Theorem 4.6 we can see how to map α_k into γ_k. In particular, the determinantal scaling of Byers [4], which for $A \in \mathbb{C}^{n\times n}$ takes $\alpha_k = |\det(X_k)^{-1/n}|$ in (6.8), yields

$$\gamma_k = |\det(Y_k)^{-1/n}| \tag{6.10}$$

in (6.9), while the spectral scaling $\alpha_k = (\rho(X_k^{-1})/\rho(X_k))^{1/2}$ of Kenney and Laub [21] yields $\gamma_k = (\rho(Y_k^{-1}Y_k^{-\star})/\rho(Y_k^{\star}Y_k))^{1/4}$. The latter acceleration parameter is suggested in [5]; it has the disadvantage of significantly increasing the cost of each iteration.

Finally, we give another example of the utility of Theorem 4.6. The Schulz iteration

$$X_{k+1} = \frac{1}{2} X_k(3I - X_k^2), \qquad X_0 = A, \tag{6.11}$$

is a member of Kenney and Laub's Padé table of iterations for $\operatorname{sign}(A)$. Applying Theorem 4.6 (or, strictly, a slightly modified version, since (6.11) is not globally convergent), we obtain the iteration

$$Y_{k+1} = \frac{1}{2} Y_k(3I - Y_k^{\star} Y_k), \qquad Y_0 = A \tag{6.12}$$

for computing W, assuming that the generalized polar decomposition $A = WS$ exists. Using a known recurrence for the residuals $I - X_k^2$ of (6.11) [1, Prop. 6.1] we find that

$$R_{k+1} = \frac{3}{4}R_k^2 + \frac{1}{4}R_k^3 \quad \text{for either } R_k = I - Y_k^{\star}Y_k \text{ or } R_k = I - Y_k Y_k^{\star}.$$

Hence a sufficient condition for the convergence of (6.12) is that the spectral radius $\rho(R_0) = \rho(I - A^{\star}A) < 1$. Iteration (6.12) was stated in [14] for the pseudo-orthogonal group, but the derivation there was ad hoc. Our derivation here reveals the full generality of the iteration.

7. Numerical properties. Key to the practical utility of the iterations we have described is their behavior in floating point arithmetic. We begin by presenting two numerical experiments in which we compute the square root of

- a random perplectic matrix $A \in \mathbb{R}^{7\times 7}$, with $\|A\|_2 = \sqrt{10} = \|A^{-1}\|_2$, generated using an algorithm of Mackey described in [18],
- a random pseudo-orthogonal matrix $A \in \mathbb{R}^{10\times 10}$, with $p = 6$, $q = 4$ and $\|A\|_2 = 10^5 = \|A^{-1}\|_2$, generated using the algorithm of Higham [14]. The matrix A is also chosen to be symmetric positive definite, to aid comparison with the theory, as we will see later.

TABLE 7.1
Results for a perplectic matrix $A \in \mathbb{R}^{7\times 7}$ with $\kappa_2(A) = 10$. Here, $\mathrm{err}(X)$ and $\mu_{\mathbb{G}}(X)$ are defined in (7.1) and (7.2).

k	Newton, (6.1)	(6.9) with $\gamma_k \equiv 1$		(6.9) with γ_k of (6.10)			Cubic, (6.4)	
	$\mathrm{err}(X_k)$	$\mathrm{err}(Y_k)$	$\mu_{\mathbb{G}}(Y_k)$	$\mathrm{err}(Y_k)$	$\mu_{\mathbb{G}}(Y_k)$	γ_k	$\mathrm{err}(Y_k)$	$\mu_{\mathbb{G}}(Y_k)$
0	1.0e+0						1.0e+0	2.5e-15
1	6.1e-1	6.1e-1	4.1e-1	6.1e-1	4.1e-1	1.4e+0	5.1e-1	8.9e-16
2	3.6e-1	3.6e-1	3.7e-1	2.5e-1	2.3e-1	1.1e+0	4.7e-2	4.4e-16
3	8.1e-2	8.1e-2	5.1e-2	2.0e-2	1.6e-2	1.0e+0	4.0e-5	4.7e-16
4	3.5e-3	3.5e-3	2.1e-3	2.3e-4	2.0e-4	1.0e+0	1.7e-14	5.3e-16
5	5.7e-6	5.7e-6	4.0e-6	1.9e-8	1.5e-8	1.0e+0	2.1e-15	4.2e-16
6	1.4e-11	1.4e-11	1.3e-11	2.0e-15	2.1e-16	1.0e+0		
7	2.2e-15	1.9e-15	1.2e-16					

TABLE 7.2
Results for a pseudo-orthogonal matrix $A \in \mathbb{R}^{10\times 10}$ with $\kappa_2(A) = 10^{10}$. Here, $\mathrm{err}(X)$ and $\mu_{\mathbb{G}}(X)$ are defined in (7.1) and (7.2).

k	Newton, (6.1)	(6.9) with $\gamma_k \equiv 1$		(6.9) with γ_k of (6.10)			Cubic, (6.4)	
	$\mathrm{err}(X_k)$	$\mathrm{err}(Y_k)$	$\mu_{\mathbb{G}}(Y_k)$	$\mathrm{err}(Y_k)$	$\mu_{\mathbb{G}}(Y_k)$	γ_k	$\mathrm{err}(Y_k)$	$\mu_{\mathbb{G}}(Y_k)$
0	3.2e+2						3.2e+2	1.4e-15
1	1.6e+2	1.6e+2	1.0e-5	1.6e+2	1.0e-5	2.0e-2	1.0e+2	7.2e-15
2	7.8e+1	7.8e+1	1.0e-5	7.4e-1	2.1e-3	3.7e-1	3.4e+1	6.0e-14
3	3.9e+1	3.9e+1	1.0e-5	1.9e-1	1.8e-4	6.5e-1	1.1e+1	5.1e-13
4	1.9e+1	1.9e+1	1.0e-5	6.0e-2	1.7e-5	8.7e-1	3.0e+0	2.9e-12
5	8.9e+0	8.9e+0	9.9e-6	4.9e-3	1.6e-6	9.8e-1	5.5e-1	4.4e-12
6	4.0e+0	4.0e+0	9.6e-6	1.2e-4	3.1e-8	1.0e+0	2.0e-2	4.1e-12
7	3.2e+1	1.6e+0	8.5e-6	3.6e-8	1.4e-11	1.0e+0	2.0e-6	4.1e-12
8	2.3e+5	4.9e-1	5.5e-6	2.1e-11	1.3e-16		2.1e-11	4.1e-12
9	4.6e+9	8.2e-2	1.5e-6					
10	2.3e+9	3.1e-3	6.1e-8					
11	1.1e+9	4.7e-6	9.5e-11					
12	5.6e+8	2.1e-11	2.4e-16					

For definitions of the perplectic and pseudo-orthogonal groups see Table 2.1. All our experiments were performed in MATLAB, for which $u \approx 1.1 \times 10^{-16}$.

Tables 7.1 and 7.2 display the behavior of the Newton iteration (6.1), the cubic iteration (6.4), iteration (6.9) without scaling, and iteration (6.9) with determinantal scaling (6.10). We report iterations up to the last one for which there was a significant decrease in the error

$$\mathrm{err}(X) = \frac{\|X - A^{1/2}\|_2}{\|A^{1/2}\|_2}. \tag{7.1}$$

We also track the departure from $\mathbb{G}$-structure of the iterates, as measured by

$$\mu_{\mathbb{G}}(X) = \frac{\|X^\star X - I\|_2}{\|X\|_2^2}; \tag{7.2}$$

see section 7.1 for justification of this measure. The next lemma gives a connection between these two quantities that applies to all the classical groups in Table 2.1.

LEMMA 7.1. *Let $A \in \mathbb{G}$, where $\mathbb{G}$ is the automorphism group of any scalar product for which M is unitary. Then for $X \in \mathbb{K}^{n\times n}$ close to $A^{1/2} \in \mathbb{G}$,*

$$\mu_{\mathbb{G}}(X) \le 2\mathrm{err}(X) + O(\mathrm{err}(X)^2). \tag{7.3}$$

Proof. Let $A \in \mathbb{G}$ and $X = A^{1/2} + E$. Then

$$\begin{aligned} X^\star X - I &= (A^{1/2})^\star (A^{1/2} + E) + E^\star A^{1/2} + E^\star E - I \\ &= A^{-1/2}E + E^\star A^{1/2} + E^\star E. \end{aligned}$$

Taking 2-norms and using (2.1) and (2.2) gives

$$\begin{aligned} \|X^\star X - I\|_2 &\le \|E\|_2(\|A^{-1/2}\|_2 + \|A^{1/2}\|_2) + \|E\|_2^2 \\ &= 2\|E\|_2\,\|A^{1/2}\|_2 + \|E\|_2^2. \end{aligned}$$

The result follows on multiplying throughout by $\|X\|_2^{-2}$ and noting that $\|X\|_2^{-2} = \|A^{1/2}\|_2^{-2} + O(\|E\|_2)$. ∎

The analysis in section 6 shows that for $A \in \mathbb{G}$ the Newton iteration (6.1) and iteration (6.9) without scaling generate precisely the same sequence, and this explains the equality of the errors in the first two columns of Tables 7.1 and 7.2 for $1 \le k \le 6$. But for $k > 6$ the computed Newton sequence diverges for the pseudo-orthogonal matrix, manifesting the well-known instability of the iteration (even for symmetric positive definite matrices). Table 7.2 shows that scaling brings a clear reduction in the number of iterations for the pseudo-orthogonal matrix and makes the scaled iteration (6.9) more efficient than the cubic iteration in this example.

The analysis of section 5 shows that the cubic structure-preserving iteration is stable, and for the classical groups it provides an estimate $(1 + \|A^{1/2}\|_2^2)u$ of the relative limiting accuracy. This fits well with the observed errors in Table 7.2, since in this example $\|A^{1/2}\|_2^2 = \|A\|_2 = 10^5$ (which follows from the fact that A is symmetric positive definite). We know from Theorem 5.5 that the unscaled iteration (6.7) is stable if the adjoint is involutory, and the same estimate of the relative limiting accuracy as for the cubic iteration is obtained for the classical groups. These findings again match the numerical results very well.

The original Newton iteration (6.1) has a Fréchet derivative map whose powers are bounded if the eigenvalues λ_i of A satisfy $\frac{1}{2}|1 - \lambda_i^{1/2}\lambda_j^{-1/2}| < 1$ for all i and j [11]. This condition is satisfied for our first test matrix but not for the second. The term on the left of this inequality also arises in Lemma 5.4 with $B = A^{1/2}$. Hence our theory predicts that the variant of (6.4) that corresponds to (5.4), in which (6.4b) is replaced by $Z_{k+1} = \frac{1}{3}Z_k[I + 8(I + 3Z_kY_k)^{-1}]$, will be unstable for the second matrix. Indeed it is, with minimum error 7.5e-3 occurring at $k = 7$, after which the errors increase; it is stable for the first matrix.

Turning to the preservation of structure, the values for $\mu_{\mathbb{G}}(Y_k)$ in the tables confirm that the cubic iteration is structure preserving. But Table 7.2 also reveals that for the pseudo-orthogonal matrix, iteration (6.9), with or without scaling, is numerically better at preserving group structure at convergence than the cubic structure-preserving iteration, by a factor 10^4. The same behavior has been observed in other examples. Partial explanation is provided by the following lemma.

LEMMA 7.2. *Assume that* $(A^\star)^\star = A$ *for all* $A \in \mathbb{K}^{n\times n}$. *If*

$$Y_{k+1} = \frac{1}{2}(Y_k + Y_k^{-\star})$$

then

$$Y_{k+1}^\star Y_{k+1} - I = \frac{1}{4}(Y_k^\star Y_k)^{-1}(Y_k^\star Y_k - I)^2.$$

Proof.

$$\begin{aligned} Y_{k+1}^\star Y_{k+1} - I &= \frac{1}{4}\big(Y_k^\star Y_k + Y_k^\star Y_k^{-\star} + (Y_k^{-\star})^\star Y_k + (Y_k^{-\star})^\star Y_k^{-\star} - 4I\big) \\ &= \frac{1}{4}(Y_k^\star Y_k + I + I + Y_k^{-1} Y_k^{-\star} - 4I) \\ &= \frac{1}{4}\big(Y_k^\star Y_k\big)^{-1}\big((Y_k^\star Y_k)^2 - 2Y_k^\star Y_k + I\big), \end{aligned}$$

which gives the result. ∎

Since Lemma 7.2 makes no assumptions about Y_k, we can think of Y_k as being an exact iterate perturbed by errors. The lemma shows that the iteration enforces quadratic convergence to the structure: an arbitrary error introduced at a particular stage can be expected to have rapidly decreasing effect on the departure from structure (though not necessarily on the error). The structure-preserving cubic iteration does not satisfy such a relation: while it automatically preserves structure, it has no mechanism for reducing a loss of structure caused by arbitrary perturbations in the iterates. However, as Lemma 7.1 shows, for any method the loss of structure is approximately bounded by the relative error, so severe loss of structure in the cubic iteration can occur only for ill-conditioned problems.

7.1. Justification of measure $\mu_{\mathbb{G}}(A)$. The measure of structure $\mu_{\mathbb{G}}$ in (7.2) was used in [15] and justified by Lemma 4.2 therein, which shows that if A has a generalized polar decomposition $A = WS$, the matrix M of the scalar product is unitary, and $\|S-I\|_2 < 1$, then $W \in \mathbb{G}$ is within relative distance approximately $\mu_{\mathbb{G}}(A)$ of A. In Theorem 7.4 below we simplify this result to assume only that $\|A^\star A - I\| < 1$ and strengthen it to apply to any consistent norm and any scalar product.

LEMMA 7.3. *Suppose that* $\operatorname{sign}(S) = I$ *and* $S^2 = I + E$, *where* $\|E\| < 1$, *for any consistent norm. Then*

$$\|S - I\| \le \frac{\|E\|}{1 + \sqrt{1 - \|E\|}} < \|E\|.$$

Proof. We will make use of the observation that if $|x| < 1$ then $(1+x)^{1/2}$ has a convergent Maclaurin series $1 + \sum_{k=1}^\infty a_k x^k$ such that $\sum_{k=1}^\infty |a_k||x|^k = 1 - \sqrt{1-x}$. Since $\operatorname{sign}(S) = I$ we have $S = (S^2)^{1/2}$ and hence $S = (I+E)^{1/2} = I + \sum_{k=1}^\infty a_k E^k$, since $\|E\| < 1$. Then

$$\begin{aligned} \|S - I\| &= \left\| \sum_{k=1}^\infty a_k E^k \right\| \le \sum_{k=1}^\infty |a_k| \|E\|^k \\ &= 1 - \sqrt{1 - \|E\|} = \frac{\|E\|}{1 + \sqrt{1 - \|E\|}} < \|E\|. \end{aligned}$$ ∎

The following theorem generalizes [12, Lem. 5.1], [14, Lem. 5.3], and [15, Lem. 4.2].

THEOREM 7.4. *Let* $\mathbb{G}$ *be the automorphism group of a scalar product. Suppose that* $A \in \mathbb{K}^{n\times n}$ *satisfies* $(A^\star)^\star = A$ *and* $\|A^\star A - I\| < 1$. *Then* A *has a generalized polar decomposition* $A = WS$ *and, for any consistent norm, the factors* W *and* S *satisfy*

$$\frac{\|A^\star A - I\|}{\|A\|(\|A^\star\| + \|W^\star\|)} \le \frac{\|A - W\|}{\|A\|} \le \frac{\|A^\star A - I\|}{\|A\|^2}\|A\|\|W\|, \tag{7.4}$$

$$\frac{\|A^\star A - I\|}{\|S\| + \|I\|} \le \|S - I\| \le \|A^\star A - I\|. \tag{7.5}$$

The inequalities (7.4) *can be rewritten as*

$$\frac{\mu_{\mathbb{G}}(A)\|A\|}{\|A^\star\| + \|W^\star\|} \le \frac{\|A - W\|}{\|A\|} \le \mu_{\mathbb{G}}(A)\|A\|\|W\|.$$

Proof. The condition $\|A^\star A - I\| < 1$ implies that the spectral radius of $A^\star A - I$ is less than 1, and hence that $A^\star A$ has no eigenvalues on $\mathbb{R}^-$. Since $(A^\star)^\star = A$, Theorem 4.1 implies that A has a (unique) generalized polar decomposition $A = WS$. Using $W^\star = W^{-1}$ and $S^\star = S$ we have

$$\begin{aligned}(A+W)^\star(A-W) &= A^\star A - A^\star W + W^\star A - W^\star W \\ &= A^\star A - S^\star W^\star W + W^\star W S - I = A^\star A - I.\end{aligned}$$

The lower bound in (7.4) follows on taking norms and using $\|(A+W)^\star\| = \|A^\star + W^\star\| \le \|A^\star\| + \|W^\star\|$.

The upper bound in (7.5) follows from Lemma 7.3, since

$$A^\star A - I = S^\star W^\star W S - I = S^2 - I. \tag{7.6}$$

The upper bound in (7.4) then follows by taking norms in $A - W = WS - W = W(S - I)$. Finally, the lower bound in (7.5) follows by writing (7.6) as $A^\star A - I = (S-I)(S+I)$ and taking norms. ∎

Note that the term $\|A^\star\|$ in the denominator of (7.4) can be replaced by $\kappa(M)\|A^T\|$ or $\kappa(M)\|A^*\|$ for bilinear forms and sesquilinear forms, respectively, and for a unitarily invariant norm both expressions are just $\|A\|$ for all the groups in Table 2.1; likewise for $\|W^\star\|$.

7.2. Conclusions on choice of method for $A^{1/2}$ when $A \in \mathbb{G}$. Our overall conclusion is that the rewritten form (6.9) of Newton's iteration, with the scaling (6.10) or perhaps some alternative, is the best iteration method for computing the square root of a matrix A in an automorphism group. This iteration

- overcomes the instability in the standard Newton iteration (6.1) and is less costly per iteration than (6.1) for the classical groups;
- is generally more efficient than the cubic structure-preserving iteration (6.4): it costs significantly less per iteration than (6.4), and (6.4) typically requires approximately the same number of iterations;
- when iterated to convergence to machine precision is likely to produce a computed result lying closer to the group than the cubic iteration (6.4) when A is ill conditioned;
- for the classical groups has half the cost per iteration of the mathematically equivalent Denman–Beavers iteration recommended in [13]. In fact, another way to derive (6.7) is to exploit the structure in the Denman–Beavers iteration that results when $A \in \mathbb{G}$.

If a structure-preserving iteration is required then an iteration from the family (6.2) can be recommended, such as the cubically convergent iteration (6.4). These iterations have the advantage that even if they are terminated well before convergence to machine precision, the result will lie in the group to approximately machine precision, though some loss of structure (no worse than that described by (7.3)) may occur for ill-conditioned problems.

Acknowledgment. We thank a referee for helpful comments and suggestions.

REFERENCES

[1] D. A. Bini, N. J. Higham, and B. Meini, *Algorithms for the Matrix pth Root*, Numerical Analysis Report No. 454, Manchester Centre for Computational Mathematics, Manchester, UK, 2004.

[2] Y. Bolshakov, C. V. M. van der Mee, A. C. M. Ran, B. Reichstein, and L. Rodman, *Extension of isometries in finite-dimensional indefinite scalar product spaces and polar decompositions*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 752–774.

[3] Y. Bolshakov, C. V. M. van der Mee, A. C. M. Ran, B. Reichstein, and L. Rodman, *Polar decompositions in finite dimensional indefinite scalar product spaces: General theory*, Linear Algebra Appl., 261 (1997), pp. 91–141.

[4] R. Byers, *Solving the algebraic Riccati equation with the matrix sign function*, Linear Algebra Appl., 85 (1987), pp. 267–279.

[5] J. R. Cardoso, C. S. Kenney, and F. Silva Leite, *Computing the square root and logarithm of a real P-orthogonal matrix*, Appl. Numer. Math., 46 (2003), pp. 173–196.

[6] H. Cartan, *Differential Calculus*, Hermann, Paris, 1971.

[7] S. H. 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.

[8] I. Gohberg, P. Lancaster, and L. Rodman, *Matrices and Indefinite Scalar Products*, Birkhäuser, Basel, Switzerland, 1983.

[9] R. E. Greene and S. G. Krantz, *Function Theory of One Complex Variable*, Wiley, New York, 1997.

[10] N. J. Higham, *Computing the polar decomposition—with applications*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 1160–1174.

[11] N. J. Higham, *Newton's method for the matrix square root*, Math. Comp., 46 (1986), pp. 537–549.

[12] N. J. Higham, *The matrix sign decomposition and its relation to the polar decomposition*, Linear Algebra Appl., 212/213 (1994), pp. 3–20.

[13] N. J. Higham, *Stable iterations for the matrix square root*, Numer. Algorithms, 15 (1997), pp. 227–242.

[14] N. J. Higham, *J-orthogonal matrices: Properties and generation*, SIAM Rev., 45 (2003), pp. 504–519.

[15] N. J. Higham, D. S. Mackey, N. Mackey, and F. Tisseur, *Computing the polar decomposition and the matrix sign decomposition in matrix groups*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 1178–1192.

[16] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.

[17] K. D. Ikramov, *Hamiltonian square roots of skew-Hamiltonian matrices revisited*, Linear Algebra Appl., 325 (2001), pp. 101–107.

[18] D. P. Jagger, *MATLAB toolbox for classical matrix groups*, M.Sc. thesis, University of Manchester, Manchester, UK, 2003.

[19] I. Kaplansky, *Algebraic polar decomposition*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 213–217.

[20] C. S. Kenney and A. J. Laub, *Rational iterative methods for the matrix sign function*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 273–291.

[21] C. S. Kenney and A. J. Laub, *On scaling Newton's method for polar decomposition and the matrix sign function*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 688–706.

[22] C. S. Kenney and A. J. Laub, *The matrix sign function*, IEEE Trans. Automat. Control, 40 (1995), pp. 1330–1348.

[23] P. Laasonen, *On the iterative solution of the matrix equation* $AX^2 - I = 0$, Math. Tables Aids Comput., 12 (1958), pp. 109–116.

[24] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, 2nd ed., Academic Press, London, 1985.

[25] D. S. Mackey, N. Mackey, and F. Tisseur, *Structured tools for structured matrices*, Electron. J. Linear Algebra, 10 (2003), pp. 106–145.

[26] D. S. Mackey, N. Mackey, and F. Tisseur, *Structured Factorizations in Scalar Product Spaces*, Numerical Analysis Report No. 432, Manchester Centre for Computational Mathematics, Manchester, UK, 2004.

[27] R. Remmert, *Theory of Complex Functions*, Springer-Verlag, Berlin, 1991.

[28] J. D. Roberts, *Linear model reduction and solution of the algebraic Riccati equation by use of the sign function*, Internat. J. Control, 32 (1980), pp. 677–687.

[29] J. T. Schwartz, *Nonlinear Functional Analysis*, Gordon and Breach, New York, 1969.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 3, pp. 878–900

QRT: A QR-BASED TRIDIAGONALIZATION ALGORITHM FOR NONSYMMETRIC MATRICES*

ROGER B. SIDJE† AND K. BURRAGE†

Abstract. The stable similarity reduction of a nonsymmetric square matrix to tridiagonal form has been a long-standing problem in numerical linear algebra. The biorthogonal Lanczos process is in principle a candidate method for this task, but in practice it is confined to sparse matrices and is restarted periodically because roundoff errors affect its three-term recurrence scheme and degrade the biorthogonality after a few steps. This adds to its vulnerability to serious breakdowns or near-breakdowns, the handling of which involves recovery strategies such as the look-ahead technique, which needs a careful implementation to produce a block-tridiagonal form with unpredictable block sizes. Other candidate methods, geared generally towards full matrices, rely on elementary similarity transformations that are prone to numerical instabilities. Such concomitant difficulties have hampered finding a satisfactory solution to the problem for either sparse or full matrices. This study focuses primarily on full matrices. After outlining earlier tridiagonalization algorithms from within a general framework, we present a new elimination technique combining orthogonal similarity transformations that are stable. We also discuss heuristics to circumvent breakdowns. Applications of this study include eigenvalue calculation and the approximation of matrix functions.

Key words. matrix reduction, nonsymmetric tridiagonalization, QR

AMS subject classifications. 15A23, 65F15, 65F25

DOI. 10.1137/040612476

1. Introduction.

The tridiagonalization of a general square matrix represents the most compact similarity reduction that can be computed directly. Attempting to reduce further (diagonalization or bidiagonalization) implies the retrieval of eigenvalues, which can only be done iteratively in general, unless the order of the matrix is under five. As yet, however, no *stable* tridiagonalization algorithm has been found. In fact we know of only one *finite*, though unstable and impractical, tridiagonalization algorithm due to George et al. [7].

While the biorthogonal Lanczos process is in principle a candidate tridiagonalization algorithm, it is generally confined to sparse matrices because rounding errors build up in its three-term recurrence scheme and degrade the biorthogonality after a few steps, making it necessary to restart periodically (though the restart may also be motivated by memory considerations in large-scale problems). The process is also vulnerable to serious breakdowns or near-breakdowns, the handling of which involves recovery strategies such as the look-ahead technique. But the look-ahead needs a careful implementation, and furthermore it produces a block-tridiagonal form with unpredictable block sizes.

Other candidate methods, geared generally towards full matrices, are not immune to serious breakdowns or near-breakdowns either, relying on elementary similarity transformations that are prone to numerical instabilities. Such concomitant difficulties have hampered finding a satisfactory solution to the problem for either sparse or full matrices. This study focuses primarily on nonsymmetric full matrices.

*Received by the editors July 29, 2004; accepted for publication (in revised form) by L. Reichel September 20, 2004; published electronically April 29, 2005.
http://www.siam.org/journals/simax/26-3/61247.html
†Department of Mathematics, Advanced Computational Modelling Centre, University of Queensland, Brisbane QLD 4072, Australia (rbs@maths.uq.edu.au, kb@maths.uq.edu.au).

Historically, interest in tridiagonalization stemmed primarily from its usefulness in reducing the cost of the LR algorithm, which predated the QR algorithm for computing the eigenvalues of a general dense matrix. However, the quest for tridiagonalization algorithms has been marred by numerical instabilities (see Wilkinson [19]). With a tridiagonalization $A = STS^{-1}$, computing an eigenpair (λ, y) of T gives the corresponding eigenpair (λ, Sy) of A. Therefore an added drawback is that eigenvectors can be contaminated when they are later retrieved by reapplying the transformations at the source of the inaccuracies.

The discovery of the QR algorithm proved very popular because it works really well, especially in conjunction with other enhancements for quick convergence (e.g., double shift) and accuracy (e.g., balancing). Tridiagonalization is unnecessary because the tridiagonal form is not preserved. The QR algorithm uses instead a preliminary orthogonal reduction to Hessenberg form for improved efficiency.

The inherent difficulties associated with tridiagonalization, together with the fact that the QR algorithm already works so well, nearly halted interest in finding stable algorithms for the reduction of a general matrix to strict tridiagonal form. But interest was rekindled by Dax and Kaniel [3], who reported that theoretical predictions are much more pessimistic than observed in practice (especially considering today's 64-bit computer architecture). Further investigation was then carried out by Geist [6], who added pivoting strategies and reported that instabilities arise at larger matrix sizes. Unfortunately, the likelihood of instabilities means that practical implementations have to anticipate them in order to remain robust and competitive [5, 6, 10, 12]. Although consolidation techniques bring some benefits, their added complexity discourages users, causing them to prefer the standard elegant QR approach with its renowned stable foundation, albeit at higher cost.

Our own interest in the problem stemmed from the computation of matrix functions [2, 15, 16]. Given a matrix A and a function f for which A is admissible (i.e., $f(A)$ is defined), the matrix function may be computed more economically as $f(A) = Sf(T)S^{-1}$, provided $A = STS^{-1}$ is a preliminary reduction to condensed form. The generic nature of the problem makes it compelling to have reduction algorithms that are useful in applications other than eigenvalue estimation.

We shall first outline a general framework for tridiagonalization algorithms. We subsequently present a new elimination technique combining orthogonal similarity transformations that are stable. We then discuss recovery techniques when serious breakdowns are encountered. We provide a roundoff error analysis. Finally, we present some numerical results and give some concluding remarks.

2. General principles of tridiagonalization.

2.1. Elementary similarity transformations. We use $\mathbb{R}$ throughout our presentation to emphasize that complex arithmetic is avoided for real data, but with minor adjustments the discussion applies to $\mathbb{C}$ as well. Let $x = (x_1, \ldots, x_n)^T$, $y = (y_1, \ldots, y_n)^T$ be vectors of $\mathbb{R}^n$, and consider the problem of finding an invertible matrix $M \in \mathbb{R}^{n \times n}$ such that

$$\left\{ \begin{array}{lcl} Mx & = & \alpha e_1, \\ y^T M^{-1} & = & \beta e_1^T, \end{array} \right. \tag{2.1}$$

where α and β are scalars to be determined and e_j is the jth column of the identity matrix of appropriate size.

LEMMA 2.1. *Assume $x_1 \neq 0$, $y_1 \neq 0$, and $y^T x \neq 0$. Then the matrices*

$$M_1 \equiv I - \frac{1}{x_1} x e_1^T + \frac{1}{y_1} e_1 y^T, \qquad M_2 \equiv I - \frac{1}{x_1} x e_1^T - \frac{1}{y_1} e_1 y^T$$

are solutions of (2.1)*, and the following hold:*

1. $M_1 x = \frac{y^T x}{y_1} e_1$; $\quad M_2 x = -\frac{y^T x}{y_1} e_1$.
2. $y^T M_1^{-1} = y_1 e_1^T$; $\quad y^T M_2^{-1} = -y_1 e_1^T$.
3. $M_1^{-1} = I - e_1 e_1^T - \frac{1}{y^T x} x (y - 2y_1 e_1)^T$; $\quad M_2^{-1} = I - e_1 e_1^T - \frac{1}{y^T x} x y^T$.
4. $\det M_1 = \frac{y^T x}{y_1 x_1}$; $\quad \det M_2 = -\det M_1$.

Proof. (1) can be verified by a straightforward multiplication; (2) and (3) follow from the Sherman–Morrison formula; finally, (4) follows from expanding the determinant. ∎

Remarks.

1. In the same spirit as other elementary transformations such as Householder, Gauss, or Gauss–Jordan, the transformation M zeroes a part of a vector. However, in contrast to those transformations, it has the special feature that its inverse M^{-1} is also a transformation targeted at a different vector.
2. Multiplying these transformations by a scalar, or more generally a diagonal matrix, preserves their basic effect. The conditions $x_1 \neq 0$ and $y_1 \neq 0$ ensure that M is defined. The condition $y^T x \neq 0$ ensures that M is invertible.
3. The inverse M^{-1} is in general a full matrix. This is, however, of limited consequence because M^{-1} is not used in isolation. What matters is its action. Notice also that if $z \in \mathbb{R}^n$, then Mz and $z^T M^{-1}$ are computed using a dot product and a gaxpy.
4. There are other matrices that satisfy Lemma 2.1. M_1 is the matrix that is often used in the literature (Geist [6] and Dongarra, Geist, and Romine [4]). It can be written as $M_1 = N_r N_c^{-1}$, where $N_r = I - \frac{1}{x_1} x e_1^T$ is the usual Gauss transformation for the row and $N_c = I - \frac{1}{y_1} e_1 y^T$ is that for the column. As we shall show in section 3, our new algorithm relies on another different type of matrix that is constructed with improved stability.

For convenience in the rest of this section we shall simply consider one case, say $M \equiv M_2$. The next illustration is a motivation for what follows. Let a matrix $A \in \mathbb{R}^{n \times n}$ be partitioned in the form

$$A = \begin{pmatrix} \delta & y^T \\ x & Z \end{pmatrix},$$

where x, $y \in \mathbb{R}^{n-1}$ for compatibility. If the assumptions of Lemma 2.1 are satisfied, a transformation M of order $n-1$ can be constructed such that

$$\begin{pmatrix} 1 & 0 \\ 0 & M \end{pmatrix} \begin{pmatrix} \delta & y^T \\ x & Z \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & M \end{pmatrix}^{-1} = \left(\begin{array}{c|ccc} \delta & \beta & 0 & \cdots & 0 \\ \hline \alpha & & & & \\ 0 & & & & \\ \vdots & & MZM^{-1} & & \\ 0 & & & & \end{array}\right).$$

The same process may be carried out on MZM^{-1} and so on. Upon termination, we end up with a tridiagonal matrix that is *similar* to the original matrix A. The method can also be used just to reduce the bandwidth of a matrix [12]. An overview of the overall procedure is outlined in the following section.

2.2. Nonsymmetric tridiagonalization. Starting with $A_0 \equiv A$ and applying the process above we obtain an updated matrix $A_{k-1} \equiv (a_{i,j}^{k-1})$ at stage $k-1$ whose pattern is

$$A_{k-1} = \left(\begin{array}{ccccc|ccccc} \delta_1 & \beta_1 & & & & & & & & \\ \alpha_1 & \delta_2 & \ddots & & & & & 0 & & \\ & \ddots & \ddots & \beta_{k-2} & & & & & & \\ & & \alpha_{k-2} & \delta_{k-1} & \beta_{k-1} & & & & & \\ \hline & & & \alpha_{k-1} & a_{k,k}^{k-1} & a_{k,k+1}^{k-1} & \cdots & a_{k,n}^{k-1} \\ & & & & a_{k+1,k}^{k-1} & a_{k+1,k+1}^{k-1} & \cdots & a_{k+1,n}^{k-1} \\ & & 0 & & \vdots & \vdots & \ddots & \vdots \\ & & & & a_{n,k}^{k-1} & a_{n,k+1}^{k-1} & \cdots & a_{n,n}^{k-1} \end{array}\right)$$

$$= \begin{pmatrix} T_{k-1} & 0 \\ 0 & A_{22}^{k-1} \end{pmatrix} + \alpha_{k-1} e_k e_{k-1}^T + \beta_{k-1} e_{k-1} e_k^T.$$

The kth transformation is then constructed as

$$M_k = \left(\begin{array}{c|cccc} I_k & & 0 & & \\ \hline & 1 & -m_{k+1,k+2} & \cdots & -m_{k+1,n} \\ & -m_{k+2,k+1} & 1 & & \\ 0 & \vdots & & \ddots & \\ & -m_{n,k+1} & & & 1 \end{array}\right)$$

$$= I - m_k e_{k+1}^T - e_{k+1} \widetilde{m}_k^T$$

where the multipliers are

$$(2.2)\quad \begin{cases} m_k = (\underbrace{0, \ldots, 0}_{k+1}, m_{k+2,k+1}, \ldots, m_{n,k+1})^T,\ m_{i,k+1} = \frac{a_{i,k}^{k-1}}{a_{k+1,k}^{k-1}},\ i = k+2, \ldots, n \\ \widetilde{m}_k = (\overbrace{0, \ldots, 0}^{k+1}, m_{k+1,k+2}, \ldots, m_{k+2,n})^T,\ m_{k+1,j} = \frac{a_{k,j}^{k-1}}{a_{k,k+1}^{k-1}},\ j = k+2, \ldots, n. \end{cases}$$

The move from stage $k-1$ to stage k is described by

$$A_k = M_k A_{k-1} M_k^{-1}.$$

The method will be of practical value if this update can be done cheaply, especially without having to manipulate M^{-1} explicitly. A handy result to that effect is summarized below, which shows how to update based on Zx/y^Tx.

LEMMA 2.2. *For the case $M \equiv M_2$, the ij-entries of $\widetilde{Z} = MZM^{-1}$ satisfy the following identities.*

Case $i = 1, j = 1$: $\widetilde{z}_{1,1} = \frac{y^T Z x}{y^T x}$.

Case $i = 1, j \neq 1$: $\widetilde{z}_{1,j} = \frac{1}{y_1}(y_j \frac{y^T Z x}{y^T x} - y^T Z e_j)$.

Case $i \neq 1, j = 1$: $\widetilde{z}_{i,1} = x_i \frac{e_1^T Z x}{x_1} - y_1 \frac{e_i^T Z x}{y^T x}$.

Case $i \neq 1, j \neq 1$: $\widetilde{z}_{i,j} = z_{i,j} - \frac{x_i}{x_1} z_{1,j} - y_j (\frac{e_i^T Z x}{y^T x} - \frac{x_i}{x_1} \frac{e_1^T Z x}{y^T x})$.

Proof. The identities come after expanding $\widetilde{z}_{i,j} = e_i^T MZM^{-1}e_j$. ∎

The method does not necessarily preserve the symmetry when the matrix is symmetric. This is of no importance since there are other techniques best suited for the symmetric case. The major concerns are rather that the algorithm may break down due to a zero pivot $a_{k,k+1}^{k-1}$ or $a_{k+1,k}^{k-1}$ in the multipliers (2.2), or it may have near-breakdowns due to small pivots that amplify the multipliers and introduce severe roundoff errors.

There is an intimate connection with the nonsymmetric (also known as biorthogonal) Lanczos process, a full description of which can be found in Golub and Van Loan [8, section 9.4.3]. The algorithm given above is equivalent to applying the biorthogonal Lanczos process to A with the starting vectors $u_1 = e_1$ and $v_1 = e_1$. In fact, any similarity tridiagonalization $A = STS^{-1}$ can always be understood as representing the biorthogonal Lanczos process with the starting vectors $u_1^T = e_1^T S^{-1}$ and $v_1 = Se_1$. In *exact* arithmetic, therefore, all tridiagonalization algorithms seeded with the same vectors are essentially equivalent, producing a tridiagonal matrix and a transformation matrix that are identical to within diagonal scaling. This is called the implicit-Q theorem, and its implications are described in detail by Parlett [14]. In particular, the desirable pairs (u_1, v_1) immune to breakdowns can be characterized in terms of Hankel determinants. It is therefore hard to tell in advance whether a pair is good, and, as with the Lanczos process, all tridiagonalization algorithms are susceptible to breakdowns.

Numerically, however, algorithms implemented in finite arithmetic may behave differently due to different stability properties. In the Lanczos process, for example, rounding errors build up rapidly in the three-term recurrence scheme and degrade the biorthogonality. It becomes unclear whether a breakdown or near-breakdown is genuine or the consequence of inaccurate intermediate computations. It is also possible for transformations of type M_1 and M_2 above to break down just because $x_1 = 0$ and/or $y_1 = 0$, or more likely they may have near-breakdowns at the neighborhood of these critical points, corrupting the ongoing tridiagonalization. Clearly, although the tridiagonalization is unique to within diagonal scaling once the starting vectors are prescribed, there can be numerical differences between algorithms, as is the case in other contexts such as in the QR decomposition which is unique but much different if computed via Modified Gram–Schmidt (MGS) or via the Classical Gram–Schmidt (CGS). Consider also QR vs. normal equations in least-squares problems or the myriad ways to get to the unique solution of a linear system. The pivoting strategy in the tridiagonalization algorithm of Geist [6] was motivated by such concerns. As we shall see later, our primary contribution is that each step of our new algorithm is nearly optimal in terms of minimizing rounding errors.

2.3. Related tridiagonalization algorithms. We will present our new algorithm in section 3. The framework that we just outlined in section 2 builds on previous works (see below). This framework bears a striking resemblance to an earlier work of Bauer [1], who showed that a class of solutions to Lemma 2.1 can be represented in the form $M = I - \tau uv^T$ and can therefore be understood as generalizations of Householder reflectors. (M_1 and M_2 above are rank-two additions and do not belong to this class unless $x = 0$ or $y = 0$.) Bauer [1] used, however, a loose terminology, defining a solution as "stable" if it exists in the neighborhood of $x_1 = 0$ and $y_1 = 0$ even though it is unstable when $y^T x \approx 0$. His transformations are also set in $\mathbb{C}^n$ and involve $\sqrt{y^T x}$ if need be, thus inducing complex arithmetic when $y^T x < 0$. Intriguingly, his work is not well publicized in the community and has gone unreferenced in other works. We

thank the anonymous referee who brought it to our attention. We believe that these general principles provide a unified and coherent approach of describing tridiagonalization algorithms. One such algorithm was described by La Budde [11], unwittingly using precisely the generalized Householder reflectors of Bauer [1]. La Budde seemed to think that his algorithm was breakdown-free. But this was promptly refuted by Parlett [13] and Wang and Gregory [18]. We cite some of the other tridiagonalization algorithms here.

ELR: This was introduced by Strachey and Francis [17]. It can be very unstable and was abandoned soon after the discovery of the QR algorithm. It was revived owing to the analysis and empirical results of Dax and Kaniel [3]. The algorithm first uses the standard Hessenberg reduction of a general matrix and then uses elementary similarity transformations to zero the terms in the upper part. One of the main weaknesses of this work is that it did not include recovery strategies.

ATOTRI: Geist [6] added a pivoting strategy to the elimination procedure in an attempt to stabilize the transformations. This proved successful in many practical problems. However, since the similarity must be preserved, there is no guarantee that the multipliers on both the column and the row will be bounded by unity. As we indicated in remark 4 above, this approach amounts to using M_1 but with pivoting. That is, the elementary transformation is $\widetilde{M} = \widetilde{N}_r(\widetilde{N}_c)^{-1}$, where $\widetilde{N}_r$ and $\widetilde{N}_c$ use Px and Py with P being a permutation matrix. Several multipliers can remain unbounded should there be no permutation P that is simultaneously suitable for $\widetilde{N}_r$ and $\widetilde{N}_c$.

BHESS: This is one of the many attempts to improve stability by reducing to a banded, as opposed to a tridiagonal, matrix. It is a variant of the elimination algorithm with pivoting in which the least stable Gauss transformations are omitted [10]. The drawback is that it produces a "trapezoidal" matrix as a compromise, i.e., an upper-Hessenberg matrix with an unfinished tridiagonalization. Thus the onus is on subsequent computations to exploit its special structure.

3. The QR-based tridiagonalization algorithm (QRT). We now describe our new tridiagonalization algorithm. It involves the following core ingredients. At each stage, it first uses a stable orthogonal similarity transformation to reduce both the column and the row. This reduces the column fully but leaves one trailing element on the row. The algorithm then finds another similarity transformation to eliminate that element. In the event of a serious breakdown, the algorithm restarts in an attempt to bypass the critical point.

3.1. The algorithm. To describe the technical aspects of the algorithm in detail, consider the partitioning

$$A = \begin{pmatrix} \delta & y^T \\ x & Z \end{pmatrix}, \tag{3.1}$$

and let

$$[x, y] = QR = Q \begin{pmatrix} \alpha & \beta \\ 0 & \gamma \\ \hline 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix}$$

be the QR decomposition of the pair $[x, y]$. Now observe that

$$\begin{pmatrix} 1 & 0 \\ 0 & Q \end{pmatrix}^T \begin{pmatrix} \delta & y^T \\ x & Z \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & Q \end{pmatrix} = \left(\begin{array}{c|ccccc} \delta & \beta & \gamma & 0 & \cdots & 0 \\ \hline \alpha & & & & & \\ 0 & & & & & \\ \vdots & & & Q^T Z Q & & \\ 0 & & & & & \end{array}\right).$$

We therefore have a configuration where the only element to be eliminated is γ. All the other unwanted elements have been eliminated in a stable manner by the Q factor. Although orthogonal transformations have been used by other tridiagonalization algorithms (e.g., for preliminary reduction to Hessenberg form), the two-sided approach that we have just illustrated above is new, and our algorithm is the first method based on this approach. This special feature forms the centerpiece of our contribution.

Now, provided γ is eliminated without significant loss of stability, we can anticipate that the overall algorithm will remain stable for many practical problems. If $|\gamma| \leq |\beta|$, it is sufficient to use an elementary similarity transformation, as we saw earlier. The following result gives a precise characterization of the elimination in advance. It also shows that it makes no difference whether we use $[x, y]$ or $[y, x]$ (i.e., the QL variant), but we shall see later that a particular choice can improve the stability of the next step.

LEMMA 3.1 (stability condition). *We have $\gamma/\beta = \tan_\theta(x, y)$, and so $|\gamma/\beta| \leq 1$ is equivalent to $|\theta| \leq \pi/4$, i.e.,*

$$|\cos_\theta(x, y)| = \frac{|x^T y|}{\|x\|_2 \|y\|_2} \geq \frac{\sqrt{2}}{2}. \tag{3.2}$$

Proof. The thin R factor in the QR decomposition of $[x, y]$ is

$$\begin{pmatrix} \alpha & \beta \\ 0 & \gamma \end{pmatrix} = \begin{pmatrix} \|x\|_2 & y^T x/\|x\|_2 \\ 0 & \|y - (y^T x/x^T x)x\|_2 \end{pmatrix}.$$

Evaluating the ratio γ/β gives the result. ∎

While bounding the multiplier by unity assists safety, the process really depends on the condition number of the similarity transformation, as our roundoff error analysis will enlighten later. Hence for the case where $|\gamma| > |\beta| > 0$ it may still be possible to apply a Gauss transformation if $|\gamma/\beta|$ does not exceed some tolerance, as done by Dax and Kaniel [3] and Geist [6], who reported that doing so is not always as bad as it seems in practice. Ideally, we would like to only use an orthogonal similarity transformation, but this is not typically possible, and our transformation attempts to come as close as we can get to one. Since the elimination step of our algorithm has only a single element to deal with, it is worth looking at the impact of a small pivot in detail. Assume that $k - 1$ steps of the tridiagonalization process have been performed, and the kth orthogonal similarity transformation has just been applied to produce the following result:

$$
\begin{aligned}
A_k^Q &= Q_k^T A_{k-1} Q_k \\
&= \left(\begin{array}{cccc|ccc|ccc}
 & & & & 0 & 0 & 0 & & & \\
 & & & & \vdots & \vdots & \vdots & & & \\
 & T_{k-1} & & & 0 & 0 & 0 & & 0 & \\
 & & & & \beta_{k-1} & 0 & 0 & & & \\
\hline
0 & \cdots & 0 & \alpha_{k-1} & \delta & \beta & \gamma & 0 & \cdots & 0 \\
0 & \cdots & 0 & 0 & \alpha & p & s & g'_{k+3} & \cdots & g'_n \\
0 & \cdots & 0 & 0 & 0 & q & t & g_{k+3} & \cdots & g_n \\
\hline
 & & & & 0 & f'_{k+3} & f_{k+3} & & & \\
 & 0 & & & \vdots & \vdots & \vdots & & H & \\
 & & & & 0 & f'_n & f_n & & &
\end{array}\right) \\
&= \left(\begin{array}{c|c|c}
T & X & 0 \\
\hline
Y & W & G \\
\hline
0 & F & H
\end{array}\right). \qquad (3.3)
\end{aligned}
$$

To zero γ, the Gauss elimination matrix is

$$
S_k = \begin{pmatrix} I_{k-1} & 0 & 0 \\ 0 & S & 0 \\ 0 & 0 & I_{n-k-2} \end{pmatrix} \in \mathbb{R}^{n\times n}, \quad S = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \mu_k \\ 0 & 0 & 1 \end{pmatrix}, \quad \mu_k = \frac{\gamma}{\beta} \equiv \mu.
$$

Using (3.3) with the fact that $SY = Y$ and $XS^{-1} = X$, we get

$$
A_k^S = S_k A_k^Q S_k^{-1} = \left(\begin{array}{c|c|c}
T & X & 0 \\
\hline
Y & SWS^{-1} & SG \\
\hline
0 & FS^{-1} & H
\end{array}\right), \qquad (3.4)
$$

where

$$
\begin{aligned}
SWS^{-1} &= \begin{pmatrix} \delta & \beta & 0 \\ \alpha & \mu q + p & -\mu^2 q + \mu(t-p) + s \\ 0 & q & -\mu q + t \end{pmatrix}, \\
SG &= \begin{pmatrix} 0 & \cdots & 0 \\ \mu g_{k+3} + g'_{k+3} & \cdots & \mu g_n + g'_n \\ g_{k+3} & \cdots & g_n \end{pmatrix}, \\
FS^{-1} &= \begin{pmatrix} 0 & \cdots & 0 \\ f'_{k+3} & \cdots & f'_n \\ -\mu f'_{k+3} + f_{k+3} & \cdots & -\mu f'_n + f_n \end{pmatrix}^T.
\end{aligned}
$$

This leads to a stable elimination of γ when $|\mu| = |\gamma/\beta| \le 1$. Unfortunately, it shows that the occurrence of a very large multiplier μ can affect a row in SG and a column in FS^{-1}. Therefore there is still a possibility of having roundoff errors that build up due to very small pivots.

In order to attempt to mitigate these difficulties, we now consider a general elimination procedure aimed at the case where $|\gamma| > |\beta| > 0$. Looking at (3.3), it can be seen that we have obtained a structure where the upcoming elimination step can be identified to the problem of reducing the smaller inner 3-by-3 block

$$
W = \begin{pmatrix} \delta & \beta & \gamma \\ \alpha & p & s \\ 0 & q & t \end{pmatrix}
$$

to tridiagonal form while using a transformation matrix that will preserve the existing zeros in the other surrounding blocks. It is not difficult to see that the corresponding transformation matrix for this must therefore have its first column and first row both equal to the first canonical basis vector (to within diagonal scaling). We can write the smaller tridiagonalization problem $\widetilde{S}W\widetilde{S}^{-1} = W'$ as

$$\left(\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & \xi_1 & \xi_2 \\ 0 & \xi_3 & \xi_4 \end{array}\right) \left(\begin{array}{c|cc} \delta & \beta & \gamma \\ \hline \alpha & p & s \\ 0 & q & t \end{array}\right) \left(\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & \xi_1 & \xi_2 \\ 0 & \xi_2 & \xi_3 \end{array}\right)^{-1} = \left(\begin{array}{c|cc} \delta & \beta' & 0 \\ \hline \alpha' & p' & s' \\ 0 & q' & t' \end{array}\right). \tag{3.5}$$

It is interesting to note that this problem does not have a solution if $\beta = 0$, just as the earlier Gauss elimination matrix was undefined in that case. This situation amounts precisely to the *serious breakdown* case in the nonsymmetric Lanczos algorithm. Indeed (3.5) is equivalent to applying the nonsymmetric Lanczos process to W with the starting vectors $u_1 = e_1$ and $v_1 = e_1$. As theory predicts [5, 7, 14], the first iteration can be taken only if we have nonzero Hankel determinants

$$\Delta_1 = u_1^T W^0 v_1 = 1 \neq 0 \quad \text{and} \quad \Delta_2 = \begin{vmatrix} u_1^T W^0 v_1 & u_1^T W^1 v_1 \\ u_1^T W^1 v_1 & u_1^T W^2 v_1 \end{vmatrix} = \alpha\beta \neq 0.$$

We will propose an heuristic for the breakdown later. Continuing for now with the premise that $|\gamma| > |\beta| > 0$, it is easily seen that a solution to (3.5) is given (to within diagonal scaling) by

$$\widetilde{S} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \tau & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad \tau = \frac{\beta}{\gamma} < 1.$$

All the quantities involved so far are computed in a stable manner. Letting

$$\widetilde{S}_k = \begin{pmatrix} I_{k-1} & 0 & 0 \\ 0 & \widetilde{S} & 0 \\ 0 & 0 & I_{n-k-2} \end{pmatrix} \in \mathbb{R}^{n\times n} \tag{3.6}$$

and using (3.3) once more with the fact that $\widetilde{S}X = X$ and $Y\widetilde{S}^{-1} = Y$, we get

$$A_k^{\widetilde{S}} = \widetilde{S}_k A_k^Q \widetilde{S}_k^{-1} = \left(\begin{array}{c|c|c} T & Y & 0 \\ \hline X & \widetilde{S}W\widetilde{S}^{-1} & \widetilde{S}G \\ \hline 0 & F\widetilde{S}^{-1} & H \end{array}\right). \tag{3.7}$$

The appearance of $\widetilde{S}^{-1}$ involves a risky division by $\tau < 1$ in the computations. This may still introduce numerical difficulties, but the potential of growth is basically confined to $F\widetilde{S}^{-1}$, and we now show how to lessen its extent. Direct calculation gives

$$\widetilde{S}W\widetilde{S}^{-1} = \begin{pmatrix} \delta & \gamma & 0 \\ \tau\alpha & q/\tau + p & -q/\tau - p + \tau s + t \\ 0 & q/\tau & -q/\tau + t \end{pmatrix}, \tag{3.8}$$

$$\widetilde{S}G = \begin{pmatrix} 0 & \cdots & 0 \\ \tau g'_{k+3} + g_{k+3} & \cdots & \tau g'_n + g_n \\ g_{k+3} & \cdots & g_n \end{pmatrix}, \tag{3.9}$$

$$F\widetilde{S}^{-1} = \begin{pmatrix} 0 & \cdots & 0 \\ f'_{k+3}/\tau & \cdots & f'_n/\tau \\ -f'_{k+3}/\tau + f_{k+3} & \cdots & -f'_n/\tau + f_n \end{pmatrix}^T. \tag{3.10}$$

This suggests that the algorithm can remain reasonably robust if we can bound most of f'_i/τ, $i = k+3, \ldots, n$. To do so, consider the earlier partitioning (3.1), and let

$$[x, y, Zx] = \widetilde{Q} \left(\begin{array}{ccc} \alpha & \beta & \rho \\ 0 & \gamma & \sigma \\ 0 & 0 & \nu \\ \hline 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{array} \right)$$

be the QR decomposition of the triplet $[x, y, Zx]$. Now observe that

$$\begin{pmatrix} 1 & 0 \\ 0 & \widetilde{Q} \end{pmatrix}^T \begin{pmatrix} \delta & y^T \\ x & Z \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \widetilde{Q} \end{pmatrix} = \begin{pmatrix} \delta & \beta & \gamma & 0 & \cdots & 0 \\ \alpha & p & \times & \times & \cdots & \times \\ 0 & q & \times & \times & \cdots & \times \\ 0 & r & \times & \times & \cdots & \times \\ 0 & 0 & \times & \times & \cdots & \times \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \times & \times & \cdots & \times \end{pmatrix}$$

with $p = \rho/\alpha, q = \sigma/\alpha, r = \nu/\alpha$. Indeed this comes from the fact that $\widetilde{Q}^T Z \widetilde{Q} e_1 = \widetilde{Q}^T Zx/\|x\|_2$. Applying this process at step k, we will therefore obtain a structure of the form

$$A_k^{\widetilde{Q}} = \widetilde{Q}_k^T A_{k-1} \widetilde{Q}_k \tag{3.11}$$

$$= \left(\begin{array}{cccc|ccc|ccc} & & & & 0 & 0 & 0 & & & \\ & & & & \vdots & \vdots & \vdots & & & \\ & & T_{k-1} & & 0 & 0 & 0 & & 0 & \\ & & & & \beta_{k-1} & 0 & 0 & & & \\ \hline 0 & \cdots & 0 & \alpha_{k-1} & \delta & \beta & \gamma & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \alpha & p & s & g'_{k+3} & \cdots & g'_n \\ 0 & \cdots & 0 & 0 & 0 & q & t & g_{k+3} & \cdots & g_n \\ \hline & & & & 0 & r & f_{k+3} & & & \\ & & 0 & & 0 & 0 & . & & H & \\ & & & & \vdots & \vdots & \vdots & & & \\ & & & & 0 & 0 & f_n & & & \end{array} \right).$$

If $|r| \leq |q|$, the term r can further be annihilated in another safe preliminary step before zeroing γ. Overall, this reorganization shows that we can significantly restrict the side effects of a large multiplier when zeroing γ in the elimination stage (3.7). Regardless the size of the problem, there can be at most six vulnerable entries at any one time: p, q, r, s, t, f_{k+3}. These are the entries where q/τ or r/τ come into play. It becomes possible to monitor them before opting for a breakdown and a recovery method. The risky entries reduce to four if $|r| \leq |q|$ and the term r is annihilated. The downside of this approach is the extra cost of Zx, but this might be preferable to other alternatives such as restarting from scratch. Moreover, the zeros that have been introduced can be exploited at the next step in a production code. If large multipliers reappear, we can alternate between the row variant and the column variant by taking the QR decomposition of $[y, x, Z^T y]$ in an attempt to diffuse the side effects evenly.

However, the trade-off now is that it brings $\widetilde{S}^{-1}$ in the transformation matrix. When a precise distinction is necessary, we shall refer to the QR of $[x, y]$ (or $[y, x]$) as the xy-QR (or yx-QR) step and to the augmented QR of $[x, y, Zx]$ (or $[y, x, Z^T y]$) as the xyz-QR (or yxz-QR) step. A reference to the QR step means either of these cases. This provides a similar distinction as with the ijk forms of loop notation.

The term q/τ, which occurs in (3.8), and the term r/τ, which will now occur in (3.10), satisfy

$$\frac{q}{\tau} = \frac{\sigma}{\alpha}\frac{\gamma}{\beta}, \quad \frac{r}{\tau} = \frac{\nu}{\alpha}\frac{\gamma}{\beta},$$

and this shows that they depend on the common quantity $\omega = \gamma/\alpha\beta$. Different values arise if we iterate with the xy- or yx-QR step. Let ω_{xy} denote the value from using the xy-QR step and ω_{yx} that of the yx-QR step. A similar reasoning as in the proof of Lemma 3.1 shows that $\omega_{xy}^2/\omega_{yx}^2 = \|x\|_2^2/\|y\|_2^2$. Hence, of the pairs $[x, y]$ and $[y, x]$, the smallest ω comes from the pair where the first vector is of smaller norm. This is how we decide whether to take a xy- or yx-QR step in practice.

In general, the unified quantity $\omega = \gamma/\alpha\beta$ highlights the relative importance of the parameters of interest in a remarkable way. If $\gamma \approx 0$, the matrix obtained after the orthogonal similarity transformation (3.3) is already tridiagonal, and so the elimination step is unnecessary. If $\alpha \approx 0$, an invariant subspace has been found, and the process can still be continued (e.g., by pivoting to eliminate β safely, also known as *deflation*), but the user may actually prefer an early termination. If $\beta \approx 0$, and $\alpha\beta$ is still small compared to γ (i.e., $|\omega|^{-1} = |\alpha\beta/\gamma| \ll 1$), there is a serious breakdown needing a full recovery method, as we shall see later. It appears therefore that, when the algorithm does continue, it does so under favorable conditions. We can choose to avoid near-breakdowns to limit the risk of introducing severe roundoff errors.

Notice that the simpler Gauss elimination matrix is a particular case of this general procedure. We use the Gauss elimination directly when $|\gamma| \leq |\beta|$, but it can be recovered here with the diagonal scaling $\mathrm{diag}(1, \frac{\gamma}{\beta}, 1)\widetilde{S}$. Also note that the transformation matrix of other solutions to (3.5) need not be necessarily triangular, though similar numerical issues arise.

For the sake of completeness, we mention another simpler but ad hoc measure which is reminiscent of diagonal scaling and thus comes with reservations. An implementation can scale (3.4) by μ^{-1} to avoid using μ, thereby preventing large numbers from being introduced as the tridiagonalization progresses. At the kth step, this scaling is summarized as

$$\mu_k^{-1} \cdots \mu_1^{-1} A_k = \mu_k^{-1} S_k Q_k^T (\cdots (\mu_1^{-1} S_1 Q_1^T A Q_1 S_1^{-1}) \cdots) Q_k S_k^{-1}.$$

Of course, the scaling factor is unity in those cases where the pivot is sufficiently large. After accumulating the scaling factors, applications can then scale back their end result when/if it is necessary to do so. A similar reasoning can be made with (3.7) using τ as scaling factors. In both cases, the danger is that not only entries of the working matrix can become quite small but that the cumulative effect of the large multipliers reappears again when unscaling the final result, suggesting that this way of doing so might not be trustworthy in general.

3.2. Breakdown and recovery. Focusing now on the more promising algorithm described earlier, it is worth noting that excessively small values of τ are often indicative of a serious breakdown requiring one to resort to recovery methods. Looking at τ alone can be too pessimistic, however. As our earlier analysis showed, the

compound quantity $\omega = \gamma/\alpha\beta$ can provide valuable insight. There are cases where a so-called *happy breakdown* may arise as well. Such cases are detected if $\alpha \approx 0$ (case of invariant subspace) or $\gamma \approx 0$ (case when the elimination step is unnecessary). The case of a serious breakdown arises when $\beta \approx 0$ after the current xy-QR step, with additionally $|\omega|^{-1} = |\alpha\beta/\gamma| \ll 1$ so that it remains unsafe to use the augmented xyz-QR step, as discussed earlier. In the Lanczos algorithm, the look-ahead technique is a popular recovery strategy for such breakdowns. However, it introduces a block-tridiagonal structure with unpredictable block sizes.

To maintain the strict tridiagonal form, it is, unfortunately, necessary to restart. This is necessary because it is not generally possible to avoid breakdowns locally. Local attempts to avoid the division by zero destroy the existing tridiagonal form. (That is why the look-ahead is the other alternative.) Dealing with breakdowns remains one of the unsatisfactory aspects of tridiagonalization algorithms. Avenues are inhibited by the tight connection to Hankel determinants and the implicit-Q theorem, as we alluded to earlier. In [19], Wilkinson suggested restarting from scratch with NAN^{-1} for some N in the hope that failure will be avoided in the modified matrix. Similarly, one can use different starting vectors, as we now describe.

We presented the QRT algorithm using $u = e_1$ and $v = e_1$ as starting vectors. Other starting vectors can be used by just applying the algorithm to the augmented matrix

$$\begin{pmatrix} 0 & u^T \\ v & A \end{pmatrix}.$$

This is tridiagonalized by the QRT algorithm as

$$\begin{pmatrix} 0 & u^T \\ v & A \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & P \end{pmatrix}^{-1} \begin{pmatrix} 0 & (u^T v/\|v\|_2)e_1^T \\ \|v\|_2 e_1 & T \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & P \end{pmatrix}$$

so that $A = P^{-1}TP$, with the first column of P^{-1} and the first row of P now

$$P^{-1}e_1 = \frac{v}{\|v\|_2}, \qquad e_1^T P = \frac{\|v\|_2}{u^T v} u^T.$$

It is always possible to choose u and v that guarantee a termination of the algorithm [7]. In general, however, whether a choice is good is not known in advance. But an interesting aspect of the principle above is that it can also be used as a recovery method. Indeed, assuming a breakdown happens at the kth step and augmenting the unfinished tridiagonalization A_{k-1}, we get

$$\begin{pmatrix} 0 & u^T \\ v & A_{k-1} \end{pmatrix} = \left(\begin{array}{c|cccc|cccc} 0 & u_1 & u_2 & \cdots & u_{k-1} & u_k & u_{k+1} & \cdots & u_n \\ \hline v_1 & \delta_1 & \beta_1 & & & & & & \\ v_2 & \alpha_1 & \delta_2 & \ddots & & & & 0 & \\ \vdots & & \ddots & \ddots & \beta_{k-2} & & & & \\ v_{k-1} & & & \alpha_{k-2} & \delta_{k-1} & \beta_{k-1} & & & \\ \hline v_k & & & & \alpha_{k-1} & a_{k,k}^{k-1} & a_{k,k+1}^{k-1} & \cdots & a_{k,n}^{k-1} \\ v_{k+1} & & & & & a_{k+1,k}^{k-1} & a_{k+1,k+1}^{k-1} & \cdots & a_{k+1,n}^{k-1} \\ \vdots & & & 0 & & \vdots & \vdots & \ddots & \vdots \\ v_n & & & & & a_{n,k}^{k-1} & a_{n,k+1}^{k-1} & \cdots & a_{n,n}^{k-1} \end{array}\right).$$

As we apply elimination steps to this augmented matrix, the existing tridiagonal form is, unfortunately, destroyed. But if we take $u = (1, r, 0, \ldots, 0)^T$ and $v = e_1$ or, alternatively, $u = e_1$ and $v = (1, r, 0, \ldots, 0)^T$, where r is a random number, we obtain what is sometimes termed "bulge chase" (see, e.g., Geist [6]). The advantage here comes from the fact that only one term has to be dealt with as the bulge is chased down to regain the tridiagonal form. This is a low-cost procedure taking $O(k)$ flops. However, in [14, section 13.3], Parlett warned that this simple recovery technique is not good enough. And indeed we observed in practice that it is not always effective in remedying breakdowns. We observed improvements when u and v had several nonzero terms, but this comes at extra cost. We could not draw from our extensive experiments a default number of nonzero terms suitable in all situations. We also noted that the bulge chase was excluded from the final Fortran code of Dongarra, Geist, and Romine [4], which was based on the work of Geist [6]. Consequently, our own implementation simply restarts by augmenting A with full vectors with components randomly chosen from a uniform distribution in the interval $(0, 1)$. We allowed only one restart in the experiments, but, as we noted before, repeatedly trying full vectors ultimately yields termination, although stability may suffer in the more difficult cases.

To detect breakdowns, we do not rely on the τ coefficients alone, as they are transient and can be too pessimistic. We instead rely on the condition number $\kappa_\infty(P) = \|P\|_\infty \|P^{-1}\|_\infty$ that can be updated incrementally from the computations. This allows us to account for the compound effects of near-breakdowns as well. In our experiments we took $\varepsilon_{\text{brk}} = 10^{-10}$ as the tolerance parameter for the breakdown; i.e., breakdown was assumed when the reciprocal of the condition number satisfies $1/\kappa_\infty(P) \le \varepsilon_{\text{brk}}$.

3.3. Roundoff error analysis. An error analysis of the elimination method in full was made by Dax and Kaniel [3]. Since the elimination step of our algorithm involves only a single element, we wish to carry out a comparative study. We leave aside the orthogonal similarity transformations. This is not much different from the approach in [3], which omitted the preliminary reduction to Hessenberg form since there are no numerical difficulties associated with orthogonal transformations. We assume the worst-case scenario of having used the augmented $\widetilde{Q}$ factor at every step. If we include roundoff errors in (3.7), the exact formulation of the kth elimination step becomes

$$A_k^{\widetilde{S}} = \widetilde{S}_k A_k^{\widetilde{Q}} \widetilde{S}_k^{-1} + \widetilde{E}_k \tag{3.12}$$

in which

$$\begin{aligned}
\widetilde{S}_k &= I + (\tau_k - 1) e_{k+1} e_{k+1}^T + e_{k+1} e_{k+2}^T, \\
\widetilde{S}_k^{-1} &= I + \left(\frac{1}{\tau_k} - 1 \right) e_{k+1} e_{k+1}^T - \frac{1}{\tau_k} e_{k+1} e_{k+2}^T, \\
\widetilde{E}_k &= \varepsilon_{pqr}^k e_{k+1}^T + \varepsilon_{stf}^k e_{k+2}^T + e_{k+1} (\varepsilon_{\alpha g}^k)^T.
\end{aligned}$$

$\widetilde{S}_k$ is the elimination matrix (3.6), and $\widetilde{E}_k$ denotes the error matrix with nonzero entries due to roundoff errors only on those positions affected by $\widetilde{S}_k$. We write $\widetilde{E}_k$ using three n-vectors for convenience. As the updating formulas (3.8)–(3.11) show, ε_{pqr}^k has only three nonzero components induced by the change of p, q, and r; ε_{stf}^k has only three nonzero components induced by the change of s, t, and f_{k+3}; and, finally,

$\varepsilon_{\alpha g}^k$ has $n-k-1$ nonzero components induced by the change of α, $g'_{k+3}, \ldots, g'_n$. Hence we have

$$e_i^T \varepsilon_{pqr}^k = 0 = e_i^T \varepsilon_{stf}^k \text{ if } i \notin \{k+1, k+2, k+3\}; \quad (\varepsilon_{\alpha g}^k)^T e_j = 0 \text{ if } j \notin \{k, k+3, \ldots, n\}.$$

Although all contributions are included in the analysis, $\varepsilon_{\alpha g}^k$ is in principle unessential since it is only induced by multiplicative terms with τ_k and by construction $\tau_k < 1$. From (3.12) the final computed result satisfies

$$A_{n-2}^{\widetilde{S}} = \widetilde{S}_{n-2} \cdots \widetilde{S}_1 A_1^{\widetilde{Q}} \widetilde{S}_1^{-1} \cdots \widetilde{S}_{n-2}^{-1} - \widetilde{E},$$
$$\widetilde{E} = \widetilde{E}_{n-2} + \widetilde{S}_{n-2} \widetilde{E}_{n-3} \widetilde{S}_{n-2}^{-1} + \sum_{k=1}^{n-4} \widetilde{S}_{n-2} \cdots \widetilde{S}_{k+1} \widetilde{E}_k \widetilde{S}_{k+1}^{-1} \cdots \widetilde{S}_{n-2}^{-1}.$$

Owing to the special pattern of $\widetilde{E}_k$, we get

$$\widetilde{E} = \widetilde{E}_{n-2} + \widetilde{S}_{n-2} \widetilde{E}_{n-3} \widetilde{S}_{n-2}^{-1} + \sum_{k=1}^{n-4} (\widetilde{S}_{k+1} + (\tau_{k+2} - 1) e_{k+3} e_{k+3}^T) \widetilde{E}_k \widetilde{S}_{k+1}^{-1} \cdots \widetilde{S}_{n-2}^{-1},$$

and using the fact that $\|\widetilde{S}_k\|_\infty = 1 + |\tau_k| \le 2$, $\|\widetilde{S}_k^{-1}\|_\infty = 2/|\tau_k|$, we obtain

$$\|\widetilde{E}\|_\infty \le \|\widetilde{E}_{n-2}\|_\infty + \sum_{k=1}^{n-3} 2 \|\widetilde{E}_k\|_\infty \|\widetilde{S}_{k+1}^{-1}\|_\infty \cdots \|\widetilde{S}_{n-2}^{-1}\|_\infty$$
$$\le 2(n-2) \max_{1 \le k \le n-2} \left(\frac{2}{|\tau_k|} \right)^{n-k-2} \max_{1 \le k \le n-2} \|\widetilde{E}_k\|_\infty. \tag{3.13}$$

The theoretical upper bound is still pessimistic, however, and the usual trade-off between speed and accuracy appears. Placing a restrictive constraint on the multipliers (e.g., $\tau_k \approx 1$) implies a growth factor, as in Gaussian elimination with partial pivoting, but a potential risk here is that recovery techniques may be triggered more often than necessary. However, this can largely be offset by the payoff from using the tridiagonal representation depending of the application. For example, Geist [6] reported a 300-by-300 eigenvalue computation on a Sun 3/280 which took 2305.14 seconds for the Hessenberg HQR method and 23.84 seconds for the tridiagonal TLR method, i.e., a hundred-fold speedup.

Further analysis suggests that our method should in general be numerically preferable over other tridiagonalization methods. As stated in Golub and Van Loan [8, eq. (7.1.11)], any similarity transformation MZM^{-1} is susceptible to roundoff errors, roughly $\varepsilon \kappa_2(M) \|Z\|_2$, where ε is the machine precision and $\kappa_2(M) = \|M\|_2 \|M^{-1}\|_2$ is the condition number. This heuristic bound implies that the safest transformation is that for which $\kappa_2(M)$ is minimum. Let M and N be transformation matrices that satisfy (2.1). There exists an invertible matrix X such that $N = XM$. It follows from (2.1) that $Xe_1 = e_1$ and $e_1^T X^{-1} = e_1^T$ (to within diagonal scaling). Take $M = M_{QRT}$ from our QRT algorithm. With the earlier notation it can be written as $M_{QRT} = SQ^T$, where $[x, y] = QR$ and $S = I + \mu e_1 e_2^T$ or $S = I + (\tau - 1) e_1 e_1^T + e_1 e_2^T = \mathrm{diag}(\mu^{-1}, 1, \ldots, 1)(I + \mu e_1 e_2^T)$. Minimizing $\kappa_2(N)$ is equivalent to minimizing $\kappa_2(XSQ^T) = \kappa_2(XS)$ over all matrices X with $Xe_1 = e_1$ and $e_1^T X^{-1} = e_1^T$. Consider now the QR factorization given by $X = YU$, where Y is orthogonal and U is upper-triangular with $Ue_1 = e_1$ and $e_1^T U = e_1^T$. The problem

becomes that of minimizing $\kappa_2(US)$ over all such U. The minimum is attained when $US = I$ or, more generally, when $US = \Pi$ with Π orthonormal. The case $US = I$ implies that $U = S^{-1}$, which is, however, inconsistent with the requirement that $e_1^T U = e_1^T$. Due to the particular structures of U and S, an effective choice has U close to the identity matrix. The case $US = \Pi$ implies that $\Pi^T US = I$, which amounts to the first case. We do not need an exact minimization of a heuristic bound. But this analysis hints at the near-optimality of our scheme with respect to minimizing roundoff errors.

The following example will illustrate the point. Let $x = (-1, 1, \ldots, (-1)^n)^T$, $y = (1, 1, \ldots, 1)^T$ of length n, with n odd to make $x^T y = 1$. Computing $M = M_2$ from Lemma 2.1 and S from the QRT scheme, we obtain

$$M = \begin{pmatrix} -1 & -1 & -1 & \cdots & -1 \\ 1 & 1 & 0 & \cdots & 0 \\ -1 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 & 0 \\ (-1)^n & 0 & \cdots & 0 & 1 \end{pmatrix}, \quad S = \begin{pmatrix} \frac{1}{\sqrt{n^2-1}} & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 1 \end{pmatrix}.$$

Hence M is stable in the Gauss sense since all its multipliers are bounded by unity. Lemma 3.1 stated the corresponding stability condition in the QRT context

$$|\cos_\theta(x, y)| = \frac{|x^T y|}{\|x\|_2 \|y\|_2} = \frac{1}{n} \not\geq \frac{\sqrt{2}}{2} = 0.707.$$

It would appear that the QRT step would not be stable in the Gauss sense, whereas methods from Lemma 2.1 would be. But for $n = 5$ we get

$$10 \approx \kappa_2(S) < \kappa_2(M) \approx 14,$$

and this shows that the QRT step is preferable, as far as the similarity transformation is concerned. Larger n gave similar observations with wider differences. For example $n = 101$ gave $\kappa_2(S) \approx 202$, whereas $\kappa_2(M) \approx 10^3$.

3.4. Pseudocode. We summarize the ideas discussed so far into a pseudocode that can be translated into a computer program. The tridiagonalization occurs in Algorithm 2 with Algorithm 1 being the driver.

ALGORITHM 1: Compute $[T, P, P_{\text{inv}}, rcond] = \boldsymbol{QRT}(A)$

$[T, P, P_{\text{inv}}, rcond] := \boldsymbol{QRTRI}(A)$;
{Attempt a recovery method if there is a breakdown}
if $rcond \leq \varepsilon_{\text{brk}}$ **then**
  Choose random u and v ;
  $[T, P, P_{\text{inv}}, rcond] := \boldsymbol{QRTRI}\left(\begin{pmatrix} 0 & u^T \\ v & A \end{pmatrix}\right)$;
  $T := T(2 : n+1, 2 : n+1)$;
  $P := P(2 : n+1, 2 : n+1)$;
  $P_{\text{inv}} := P_{\text{inv}}(2 : n+1, 2 : n+1)$;
endif

Note in the pseudocode that a quantity $\theta \approx 0$ if $|\theta| \leq \varepsilon_{\text{zero}}$. Our MATLAB implementation used the drop tolerance $\varepsilon_{\text{zero}} = 10^{-7}$. Note also that each iteration of the pseudocode begins by deciding whether to take a xy- or yx-QR step. Any subsequent action then uses the appropriate indices, depending on the step retained.

Details are omitted in the pseudocode for readability. When there is breakdown, the control is passed back to the driver routine to possibly initiate a recovery attempt. An implementation can choose to exit with the last good values before the breakdown in case the user wants them, albeit they represent a partial decomposition.

ALGORITHM 2: Compute $[T, P, P_{\text{inv}}, rcond] = \boldsymbol{QRTRI}(A)$

$P := I$; $P_{\text{inv}} := I$; $T := A$;
for $k := 1 : n-2$ **do**
  $x := T(k+1:n, k)$; $y := T(k, k+1:n)^T$;
  {Decide whether to use $[x, y]$ or $[y, x]$}
  if $\|x\|_2 \le \|y\|_2$ **then**
    $[Q, R] := QR(x, y)$;
  else
    $[Q, R] := QR(y, x)$;
  endif
  $\alpha = R(1,1)$; $\beta = R(1,2)$; $\gamma = R(2,2)$;
  {Use the simple xy- or yx-QR step if no xyz- or yxz-QR step is needed}
  if $\alpha \approx 0$ **or** $\gamma \approx 0$ **or** $|\beta| \ge |\gamma|$ **then**
    $T := \begin{pmatrix} I_k & 0 \\ 0 & Q^T \end{pmatrix} T \begin{pmatrix} I_k & 0 \\ 0 & Q \end{pmatrix}$; $P := \begin{pmatrix} I_k & 0 \\ 0 & Q^T \end{pmatrix} P$; $P_{\text{inv}} := P_{\text{inv}} \begin{pmatrix} I_k & 0 \\ 0 & Q \end{pmatrix}$
  endif
  {Move on to the next step if no elimination is necessary}
  if $\gamma \approx 0$ **continue** ;
  {Deflation when we have an invariant subspace}
  if $\alpha \approx 0$ **then**
    • apply Gauss elimination with pivoting to eliminate γ or β in T
    • update the transformation matrix P and its inverse P_{inv}
    continue ;
  endif
  {Use the simple Gauss elimination if possible}
  if $|\beta| \ge |\gamma|$ **then**
    • apply Gauss elimination to eliminate γ in T
    • update the transformation matrix P and its inverse P_{inv}
    continue ;
  endif
  {Use the xyz- or yxz-QR elimination if possible}
  if $\beta \approx 0$ **then**
    {serious breakdown}
    set $rcond := 0$;
  else
    • apply the extended xyz- or yxz-QR step
    • eliminate the r term if possible in T — see the discussion following (3.11)
    • eliminate γ in T
    • update the transformation matrix P and its inverse P_{inv}
    • compute $rcond := 1/\|P\|_\infty \|P_{\text{inv}}\|_\infty$, the reciprocal of the condition number
  endif
  {Exit if there is a breakdown}
  if $rcond \le \varepsilon_{\text{brk}}$ **return** ;
endfor

3.5. A breakdown-free variant. We outline here a modified variant useful in certain applications. This variant avoids serious breakdowns at the trade-off of not producing a strict tridiagonal form. Consequently, we call it the breakdown-free QRT (BFQRT). There are applications where a strict tridiagonal form (or a form with

bandwidth fourth or more) is not essential. But having as many zeros as possible is key to efficiency because floating-point operations involving zero elements can be avoided. This can be seen, for example, in Nikolajsen [12], where skipping null elements in the Laguerre eigensolver resulted in a marked speedup over the QR algorithm.

The BFQRT variant consists of omitting the elimination steps that would normally trigger recovery attempts. A similar strategy is used in BHESS [10] and Nikolajsen [12]. However, their approach gives a "trapezoidal" matrix of increasing bandwidth, whereas our approach reduces the density further by retaining a tridiagonal matrix but with occasional rows on the upper part. These rows appear where the elimination steps have not been applied. The pseudocode for this looks similar to Algorithm 2, except that we use only the xy- or xyz-QR steps and do not alternate with the yx- or yxz-QR steps. Another difference is that if $\beta \approx 0$, we just move on to the next step. We also use the updated *rcond* merely to decide whether to revert to the last good values before proceeding with the next step. Below are examples of patterns that BFQRT may produce in a 7-by-7 case:

$$\begin{pmatrix} \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & & & & \\ & \times & \times & \times & \times & \times & \times \\ & & \times & \times & \times & & \\ & & & \times & \times & \times & \\ & & & & \times & \times & \times \\ & & & & & \times & \times \end{pmatrix}, \quad \begin{pmatrix} \times & \times & & & & & \\ \times & \times & \times & \times & \times & \times & \times \\ & \times & \times & \times & & & \\ & & \times & \times & \times & & \\ & & & \times & \times & \times & \\ & & & & \times & \times & \times \\ & & & & & \times & \times \end{pmatrix}.$$

Note that in practice it is not necessary to apply a xy-QR step on rows set to be filled again. We can use a simple Hessenberg step there and move on to the next iteration.

4. Numerical experiments. We report some numerical examples using an exploratory MATLAB implementation on a Sun4u Sparc Workstation. Given a matrix A, we apply the QRT algorithm to compute $T = PAP^{-1}$, where T is tridiagonal and P is the similarity transformation matrix.

We compare our method with the ATOTRI Fortran code of Dongarra, Geist, and Romine [4] and Geist [6]. To this end, we implemented a MEX interface to invoke the native Fortran code of ATOTRI from within MATLAB. The comparison is based therefore on their original Fortran implementation available in the TOMS directory at netlib.org.

In the first set of examples, we also use $[L, U] = \mathrm{lu}(A)$ in MATLAB to compute the LU decomposition with partial pivoting. We report $\|U\|_\infty$, which gives insight into the growth factor that would arise with the Gauss elimination procedure itself. The following statistics (as computed by MATLAB) are given to assist in the evaluation of the results:

n	order of the matrix A
X_{eig}	eigenvectors of A, as computed by $[X, D] = \mathrm{eig}(A)$ in MATLAB
$\|U\|_\infty$	L-∞ norm, indicator of the growth in the LU decomposition of A
$\kappa_2(P)$	condition number of the matrix P, $\kappa_2(P) = \|P\|_2 \|P^{-1}\|_2$

4.1. GFPP examples. Results are shown on Table 1. The matrices are generated using the function called `gfpp` in Higham's testsuite [9]. This function generates a matrix that has the effect of attaining the maximal growth factor in Gaussian elimination with partial pivoting. We use `gfpp(n, c)`, which sets all the multipliers to c and gives a growth factor $(1 + c)^{n-1}$.

TABLE 1
GFPP examples.

GFPP Problem	$\|A\|_2$	$\kappa_2(X_{\text{eig}})$	$\kappa_2(P)$	$\frac{\|A-P^{-1}TP\|_2}{\|A\|_2}$	$\|U\|_\infty$	$\frac{\|A-LU\|_2}{\|A\|_2}$
$n = 50$, $c = 0.3$	1.10E+01	3.42E+00	E+01	E−15	E+05	E−12
$n = 100$, $c = 0.3$	2.05E+01	4.74E+00	E+01	E−15	E+11	E−06
$n = 200$, $c = 0.3$	1.04E+07	1.06E+07	E+02	E−15	E+22	E−02

Although this problem clearly affects the LU algorithm as n increases, it is handled well by the tridiagonalization method. This supports the observation made by Dax and Kaniel [3] that tridiagonalization methods are not necessarily doomed to fail on practical problems. In the same spirit that LU can fail but is widely used nonetheless, cheap elimination methods can be tried first before resorting to other robust (but expensive) alternatives to compute eigenvalues.

4.2. EigTool examples. Results appear on Table 2 and Figure 1. Most of the matrices in the EigTool set [20] are notoriously pathological. They are specifically aimed at showcasing the importance of pseudospectra analysis, and so each eigensystem is very sensitive to small perturbations. We refer the reader to EigTool [20] for further details about these problems. The QRT tridiagonalization is successful for most cases but suffers from serious breakdowns in some cases. The column with label *err* gives an error exit status. A value of 0 means that the algorithm completed all the steps. A value in the form $(k_1)k_2$ means that the algorithm encountered a serious breakdown at the k_1th step and the conservative recovery technique discussed earlier in section 3.2 was applied. A null k_2 means that the recovery was successful. Otherwise it means that the recovery itself failed at the k_2th step. We allocated a similar column for ATOTRI, but as our analysis will show, its error exit status is not entirely reliable.

TABLE 2
EigTool examples.

			QRT			ATOTRI		
Problem	$\|A\|_2$	$\kappa_2(X_{\text{eig}})$	err	$\kappa_2(P)$	$\frac{\|A-P^{-1}TP\|_2}{\|A\|_2}$	err	$\kappa_2(P)$	$\frac{\|A-P^{-1}TP\|_2}{\|A\|_2}$
hatano 50x50	2.93E+00	E+07	0	1	0	0	1	0
demmel 50x50	3.19E+04	Inf	0	E+04	E−15	0	E+08	E−13
gallery3 3x3	8.18E+02	E+03	0	7.55	E−16	0	E+01	E−17
gallery5 5x5	1.01E+05	E+11	0	E+07	E−15	0	E+08	E−13
godunov 7x7	4.32E+03	E+14	0	E+03	E−14	0	E+04	E−15
convdiff 49x49	1.02E+04	E+12	0	E+04	E−15	0	E+04	E−14
chebspec 49x49	1.32E+03	E+13	0	E+02	E−13	0	E+03	E−13
kahan 50x50	5.76E+00	E+12	0	E+01	E−15	0	E+02	E−15
sparserandom 50x50	3.28E+00	E+01	0	E+04	E−11	0	E+04	E−11
random 50x50	1.95E+00	E+01	0	E+03	E−14	0	E+02	E−13
boeing 55x55	1.69E+07	E+06	(17)0	E+07	E−15	–	−−	−−
twisted 50x50	2.74E+00	E+05	(37)0	E+07	E−11	0	E+10	E−06
frank 50x50	6.73E+02	E+10	(7)0	E+07	E−12	0	E+11	E−02
grcar 50x50	3.23E+00	E+08	(15)0	E+08	E−09	0	E+14	E−02
companion 50x50	4.59E+64	E+63	(1)0	E+04	E−16	–	−−	−−
markov 55x55	1.18E+00	E+02	(26)0	E+04	E−10	0	E+10	E−03
randomtri 50x50	1.64E+00	E+18	(25)25	E+07	E−10	–	*E*+47	*E*+16
riffle 50x50	2.36E+00	E+44	(8)11	E+08	E−10	–	*E*+35	*E*+05

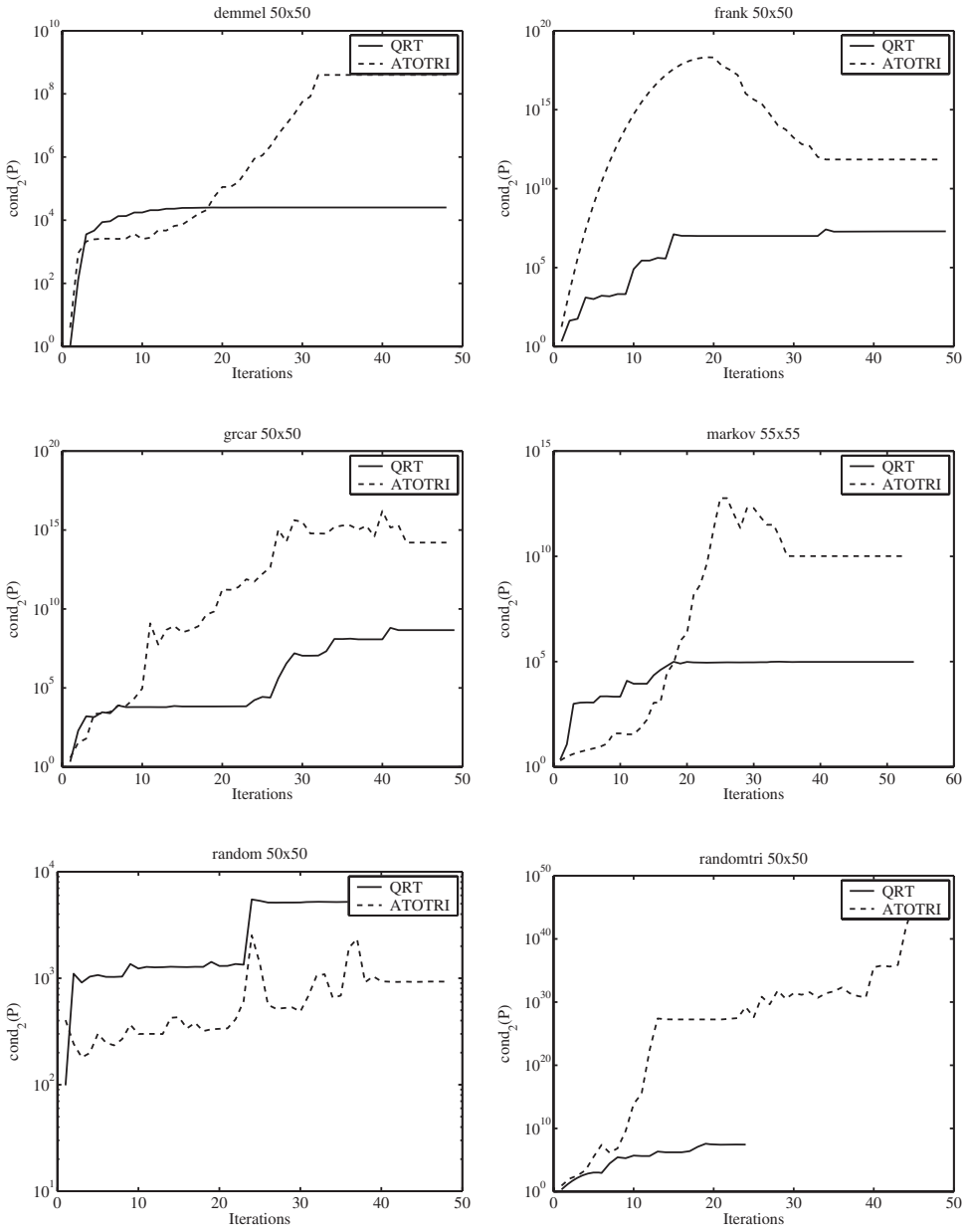


FIG. 1. *History of the condition number of the transformation matrix,* $\mathrm{cond}_2(P_k) = \|P_k\|_2\|P_k^{-1}\|_2$, *during the reduction of a few representative matrices from EigTool.*

Recall that $\frac{\|A-P^{-1}TP\|_2}{\|A\|_2}$ is bound to have rounding errors of order $\varepsilon\kappa_2(P)$, where ε is the machine precision, which is about 10^{-16} on the Sun4u Sparc Workstation where we conduct the experiments. We can make the following main observations:

- The *hatano* matrix is already tridiagonal and should be left untouched because the methods used $u = e_1$ and $v = e_1$ as default starting vectors. Thus this matrix served as an identity test for the codes.
- It is clear from the table that QRT is more accurate than ATOTRI in gen-

eral. The plots in Figure 1 depict the history of the condition number of the transformation matrix in various examples. There are occasional cases such as *random* where ATOTRI looks better. But even in those cases the relative error of QRT is as good as or better than ATOTRI, as seen on the main results on Table 2. In general, therefore, the transformation matrix produced by QRT tends to have the smallest condition number, leading to a smaller relative error. This agrees with the roundoff error analysis.

- The behavior of ATOTRI is disturbing in a number of pathological cases where its computed solution is seriously contaminated by roundoff errors, but the user is not given any warning. We use a dash (−) on Table 2 to draw the attention of the reader in those cases. The code actually returns an error exit status *err* of 0 that can mislead the user into thinking that the result is reliable when in fact there has been a total loss of accuracy. The *companion* example gave huge values. Another dramatic example was the *boeing* matrix that gave a transformation matrix for which the singular value decomposition to compute its condition number failed. Other examples are *riffle* and *randomtri*. As the history of *randomtri* in Figure 1 shows, QRT stopped at some point after reporting that its recovery attempt failed. But ATOTRI continued with meaningless data. Looking at the Fortran code of ATOTRI, we noted that it does not account for near-breakdowns. It detects the breakdown only if the inner product $x^T y = 0$. Other ramifications can be seen in the *frank* example: the condition number grows exaggeratedly before decaying, with the effect of corrupting the rest of the computations in a way not made apparent to the user. Such examples justify the careful attention for a more reliable breakdown criteria, as done by QRT.
- In the successful cases, a few problems (those for which *err* is in the form $(k_1)k_2$) needed recovery from breakdown. Recall that our recovery method consists of restarting with random vectors. Restarting was allowed only once.
- It can be seen that failure often arises because the conditioning of the eigensystem is simply too large compared with the norm of A. This is the case for the *randomtri* and *riffle* examples. A breakdown that happens very late in the tridiagonalization is suggestive of a critical choice of starting vectors. It is worth nothing that the results remain meaningful because they represent an unfinished tridiagonalization, which can still be useful, as the error bound shows. In those cases, it should be understood that $T = A_{k-1}$ for some k and is not really tridiagonal. See, for example, (3.11). Recall that the tridiagonalization is not an end in itself. When k is close to n, the remaining block can be reduced to Hessenberg form, and/or subsequent computations can take advantage of this nearly tridiagonal structure.

In other less pathological problems not reported here, QRT had a similar pattern of encouraging results. Overall, therefore, this algorithm was generally successful.

4.3. Eispack examples. We also applied our algorithm to matrices in the testsuite of Eispack. Results are displayed in Table 3 and Figure 2. This testsuite consists of 35 small matrices (none exceeding 20×20) that were thoughtfully designed to exercise the general purpose eigensolvers in Eispack. As in EigTool, the matrices are pathological with defective and/or derogatory cases. The examples do not appear as challenging as the EigTool examples, and we note that both algorithms were successful on all of the problems, and the accuracy remains very good. There are cases where recovery is needed at the very first step, suggesting that $u = e_1$ and $v = e_1$ are not

TABLE 3
Eispack examples.

			QRT			ATOTRI		
Problem	$\|A\|_2$	$\kappa_2(X_{\text{eig}})$	err	$\kappa_2(P)$	$\frac{\|A-P^{-1}TP\|_2}{\|A\|_2}$	err	$\kappa_2(P)$	$\frac{\|A-P^{-1}TP\|_2}{\|A\|_2}$
1: 8x8	1.02E+03	1	0	1	E−15	0	2.05	E−16
2: 6x6	2.66E+09	5.73	0	1.69	E−16	0	3.28	E−16
3: 5x5	4.55E+01	E+08	0	E+02	E−16	0	E+02	E−15
4: 12x12	6.34E+01	1	0	1	E−16	0	1	E−16
5: 10x10	1.92E+08	E+02	0	E+02	E−16	0	E+02	E−15
6: 15x15	6.68E+06	E+01	0	E+01	E−15	0	E+01	E−15
7: 19x19	5.96E+05	E+01	0	E+03	E−14	0	E+03	E−13
8: 6x6	0	1	0	1	NaN	0	1	NaN
9: 6x6	5.58E+01	E+11	0	E+01	E−15	0	E+01	E−16
10: 6x6	1.67E+06	E+02	(2)0	E+01	E−15	(2)0	E+02	E−15
11: 5x5	2.41E+01	2.45	0	7.66	E−15	0	9.34	E−16
12: 5x5	1.93E+01	3.40	(1)0	6.32	E−16	(1)0	6.64	0
13: 5x5	1.93E+01	3.27	(1)0	E+01	E−14	(1)0	E+02	E−14
14: 5x5	2.07E+01	2.69	0	E+02	E−15	0	E+02	E−14
15: 5x5	2.07E+01	2.72	0	E+01	E−15	0	E+01	0
16: 3x3	1.80E+01	Inf	0	1	0	0	1	0
17: 3x3	1.07E+02	Inf	0	1	0	0	1	0
18: 3x3	1.06E+01	Inf	0	1	0	0	1	0
19: 4x4	1.26E+02	E+11	(1)0	2.49	E−17	(1)0	2.43	E−18
20: 3x3	1.00E+01	1	(1)0	5.17	E−15	(1)0	5.04	E−16
21: 4x4	1.00E+01	1	(1)0	2.75	E−15	(1)0	3.20	E−16
22: 5x5	1.00E+01	1	(1)0	E+01	E−14	(1)0	E+02	E−14
23: 6x6	1.00E+01	1	(1)0	E+01	E−14	(1)0	E+01	E−14
24: 8x8	1.00E+09	E+07	0	E+02	E−14	0	E+01	E−14
25: 4x4	7.01E+01	2.25	0	E+02	E−15	0	E+02	0
26: 3x3	7.12E+01	6.61	0	2.62	0	0	2.62	0
27: 4x4	4.35E+01	E+08	0	5.31	E−16	0	6.81	E−17
28: 4x4	1.23E+02	5.98	0	4.73	E−16	0	6.26	E−16
29: 6x6	7.28E+01	E+01	0	E+01	E−15	0	E+01	E−16
30: 6x6	1.93E+02	E+01	0	E+01	E−14	0	E+02	E−15
31: 8x8	2.37E+01	1.80	0	E+01	E−15	0	E+01	E−15
32: 4x4	1.78E+02	E+12	0	3.94	E−16	0	4.67	E−16
33: 6x6	1.46E+02	E+12	0	E+01	E−15	0	E+01	E−16
34: 8x8	3.12E+02	E+11	0	E+02	E−14	0	E+03	E−14
35: 10x10	1.08E+02	E+10	0	E+03	E−13	(5)0	E+02	E−15

suitable as default starting vectors there. Notice that the matrix in problem 8 is zero and that this is why the relative error is *NaN* (Not-a-Number).

5. Conclusion. We have described a promising algorithm for the tridiagonalization of nonsymmetric matrices. The algorithm primarily involves two stable Householder transformations per step and is twice as expensive as the symmetric tridiagonalization. The robust QR step provides a solid foundation to the proposed algorithm. There is still the possibility of suffering from the effect of a large multiplier, but we showed how to restrict risky roundoff errors to at most six entries irrespective of the size of the matrix. This suggests that the algorithm may be of assistance in a wide class of practical problems where a preliminary tridiagonalization is useful. Recovery techniques were discussed in the case where a serious breakdown happens or when a small pivot is rejected. A breakdown-free variant was described with the trade-off of not producing a strict tridiagonal form. Largely successful numerical experiments were conducted using a conservative restarting criteria to ascertain the robustness of the method. A comparison was made with a previous tridiagonalization algorithm of

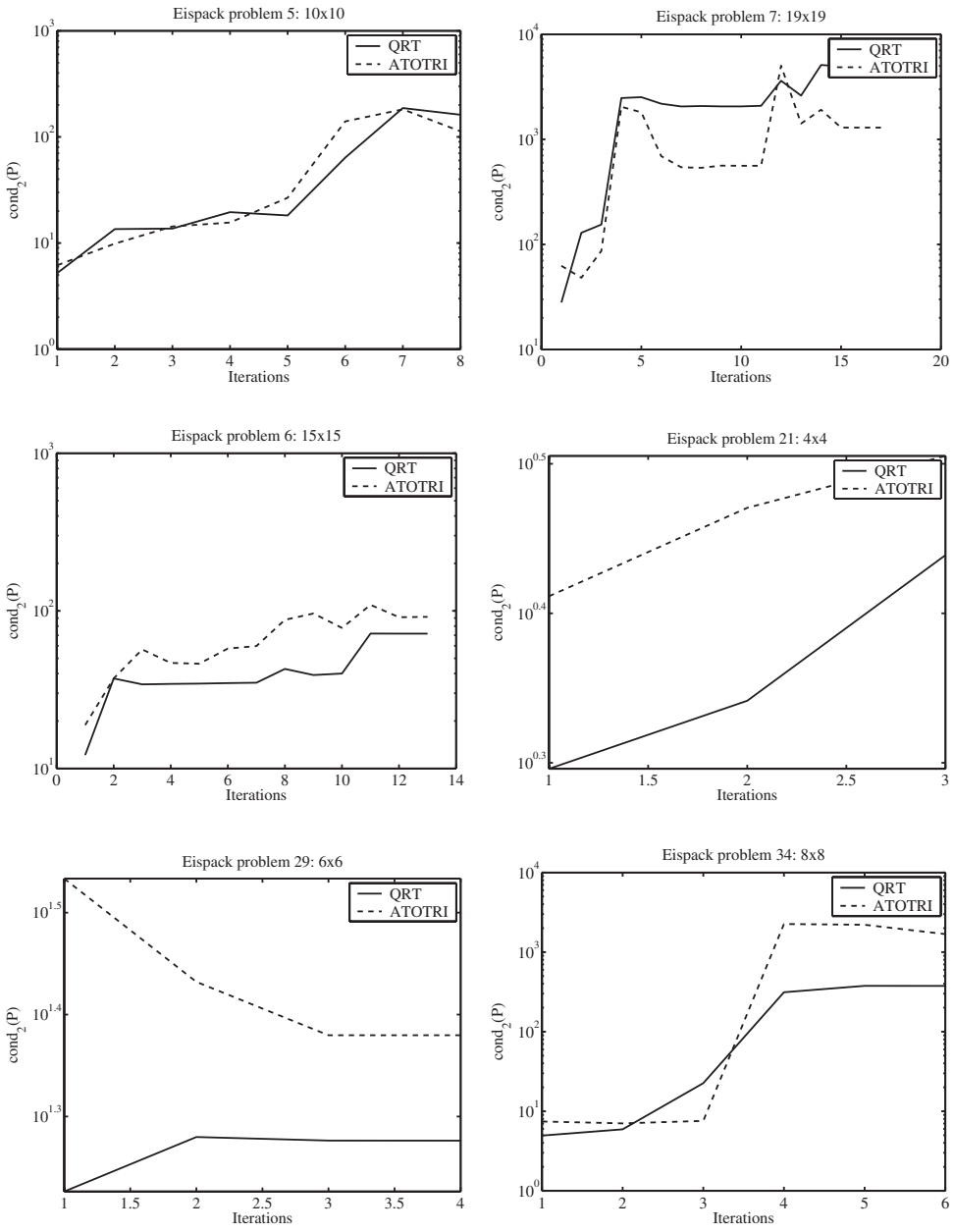


FIG. 2. *History of the condition number of the transformation matrix,* $\text{cond}_2(P_k) = \|P_k\|_2\|P_k^{-1}\|_2$, *during the reduction of a few representative matrices from Eispack.*

Dongarra, Geist, and Romine [4] and Geist [6], and it shows that our algorithm is generally more robust and reliable. A roundoff error analysis suggests that our method should in general be numerically preferable over other tridiagonalization methods because it is nearly optimal in minimizing roundoff errors.

Acknowledgment. We would like to thank Prof. Nick Trefethen for his comments on drafts of this paper.

REFERENCES

[1] F. L. Bauer, *Sequential reduction to tridiagonal form*, J. Soc. Indust. Appl. Math., 7 (1959), pp. 107–113.

[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] A. Dax and S. Kaniel, *The ELR method for computing the eigenvalues of a general matrix*, SIAM J. Numer. Anal., 18 (1981), pp. 597–605.

[4] J. J. Dongarra, G. A. Geist, and C. H. Romine, *Algorithm* 710: *FORTRAN subroutines for computing the eigenvalues and eigenvectors of a general matrix by reduction to general tridiagonal form*, ACM Trans. Math. Software, 18 (1992), pp. 392–400.

[5] R. W. Freund, M. H. Gutknecht, and N. M. Nachtigal, *An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices*, SIAM J. Sci. Comput., 14 (1993), pp. 137–158.

[6] G. A. Geist, *Reduction of a general matrix to tridiagonal form*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 362–373.

[7] A. George, K. Ikramov, A. N. Krivoshapova, and W.-P. Tang, *A finite procedure for the tridiagonalization of a general matrix*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 377–387.

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore,1996.

[9] N. J. Higham, *The Test Matrix Toolbox for MATLAB (Version* 3.0*)*, Numerical Analysis Report 276, Department of Mathematics, University of Manchester, Manchester, UK, 1995.

[10] G. W. Howell, *Efficient computation of eigenvalues of randomly generated matrices*, Appl. Math. Comput., 66 (1994), pp. 9–24.

[11] C. D. La Budde, *The reduction of an arbitrary real square matrix to tridiagonal form using similarity transformations*, Math. Comp., 17 (1963), pp. 433–437.

[12] J. L. Nikolajsen, *An improved Laguerre eigensolver for unsymmetric matrices*, SIAM J. Sci. Comp., 22 (2000), pp. 822–834.

[13] B. N. Parlett, *A note on La Budde's algorithm*, Math. Comp., 18 (1964), pp. 505–506.

[14] B. N. Parlett, *Reduction to tridiagonal form and minimal realizations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 567–593.

[15] R. B. Sidje, *EXPOKIT. A software package for computing matrix exponentials*, ACM Trans. Math. Software, 24 (1998), pp. 130–156.

[16] R. B. Sidje, K. Burrage, and B. Philippe, *An augmented Lanczos algorithm for the efficient computation of a dot-product of a function of a large sparse symmetric matrix*, in Proceedings of the International Conference on Computational Science, Lecture Notes in Comput. Sci. 2659, P. M. A. Sloot et al. eds., Springer-Verlag, Berlin, 2003, pp. 693–704.

[17] C. Strachey and J. G. F. Francis, *The reduction of a matrix to codiagonal form by eliminations*, Comput. J., 4 (1961), pp. 168–176.

[18] H. H. Wang and R. T. Gregory, *On the reduction of an arbitrary real square matrix to tridiagonal form*, Math. Comp., 18 (1964), pp. 501–505.

[19] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, England, 1965.

[20] T. G. Wright, *EigTool Software Package*; http://web.comlab.ox.ac.uk/pseudospectra/eigtool/.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 4, pp. 901–907

INVARIANT SUBSPACES OF SKEW-ADJOINT MATRICES IN SKEW-SYMMETRIC INNER PRODUCTS*

LEIBA RODMAN†

Abstract. It is proved that, for a real matrix which is skew-adjoint with respect to a skew-symmetric inner product, every given neutral invariant subspace is contained in an invariant subspace which is also maximal semidefinite with respect to an associate symmetric bilinear form. Applications are given to the existence of solutions of continuous and discrete algebraic Riccati equations, with the property that the ranks of skew-symmetric parts of the solutions have a fixed upper bound. As a particular case, a known basic result concerning symmetric solutions of the Riccati equations is recovered.

Key words. skew-symmetric inner product, skew-adjoint matrix, continuous algebraic Riccati equation, discrete algebraic Riccati equation

AMS subject classifications. 15A57, 15A63, 93B99

DOI. 10.1137/S0895479804439213

1. Introduction. All matrices are assumed to be real. Denote by $\mathbb{R}^{p\times q}$ the vector space (algebra if $p = q$) of $p \times q$ real matrices. A skew-symmetric matrix, which is allowed to be singular, $H \in \mathbb{R}^{m\times m}$ induces a skew-symmetric inner product $[x, y]_H = y^T Hx$, $x, y \in \mathbb{R}^m$. (The superscript ^{T} stands for the transpose.) A matrix $A \in \mathbb{R}^{m\times m}$ is called *H-skew-adjoint* if $[Ax, y]_H = -[x, Ay]_H$ for all $x, y \in \mathbb{R}^m$, in other words, if the matrix HA is symmetric. If $\left(\begin{smallmatrix} 0 & I \\ -I & 0 \end{smallmatrix}\right)$, then H-skew-adjoint matrices are often called Hamiltonian (see, for example, [7]). Pairs of matrices (A, H), where H is skew-symmetric and A is H-skew-adjoint, play a key role in several significant problems of applied analysis, in particular Riccati equations (which are ubiquitous in systems and control); see, e.g., the books [1, 10, 16, 17], gyroscopic vibrating systems [4, 11], Hamiltonian systems, and transfer functions with symmetries and their factorizations (see, e.g., [2, 8, 14, 19]).

In this paper we study invariant subspaces of H-skew-adjoint matrices that have neutrality or definitiveness properties with respect to the skew-symmetric inner product induced by H. The study is motivated by applications to algebraic Riccati equations to be presented in section 3. Since H is skew-symmetric, and therefore $[x, x]_H = 0$ for every vector $x \in \mathbb{R}^m$, to formulate the definiteness properties we will use the bilinear form defined by the symmetric matrix HA.

To state the main results we recall several definitions and introduce some notation. Throughout this discussion we fix a skew-symmetric $H \in \mathbb{R}^{m\times m}$ and an H-skew-adjoint $A \in \mathbb{R}^{m\times m}$. A subspace $\mathcal{M} \subseteq \mathbb{R}^m$ is called *H-neutral* if $[x, y]_H = 0$ for all $x, y \in \mathcal{M}$. The maximal dimension of an H-neutral subspace is easily seen to be $\dim(\operatorname{Ker} H) + \frac{1}{2}(\operatorname{rank} H)$. A subspace $\mathcal{M} \subseteq \mathbb{R}^m$ is called *HA-nonnegative* (resp., *HA-nonpositive*) if $x^T HAx \geq 0$ (resp., $x^T HAx \leq 0$) for all $x \in \mathcal{M}$. An HA-nonnegative subspace is called *maximal HA-nonnegative* if it is not properly contained

*Received by the editors December 28, 2003; accepted for publication (in revised form) by V. Mehrmann August 24, 2004; published electronically May 6, 2005. This research was partially supported by NSF grant DMS-9988579.

http://www.siam.org/journals/simax/26-4/43921.html

†College of William and Mary, Department of Mathematics, P.O. Box 8795, Williamsburg, VA 23187-8795 (lxrodm@math.wm.edu).

in any larger HA-nonnegative subspace; maximal HA-nonpositive subspaces are defined analogously. It is well known that an HA-nonnegative (resp., HA-nonpositive) subspace $\mathcal{M}$ is maximal if and only if

$$\dim \mathcal{M} = \nu_+(HA) + \nu_0(HA) \quad (\text{resp.}, \quad \dim \mathcal{M} = \nu_-(HA) + \nu_0(HA)),$$

where we denote by $\nu_+(G)$, $\nu_-(G)$, and $\nu_0(G)$ the numbers of positive, negative, and zero eigenvalues, respectively, of a symmetric matrix G (counted with multiplicities).

THEOREM 1.1. *Let $H, A \in \mathbb{R}^{m\times m}$ be such that H is skew-symmetric and A is H-skew-adjoint. Let $\mathcal{N} \subseteq \mathbb{R}^m$ be an A-invariant H-neutral subspace. Then there exist A-invariant subspaces $\mathcal{L}_+$ and $\mathcal{L}_-$ such that each of them contains $\mathcal{N}$ and $\mathcal{L}_+$ is maximal HA-nonnegative, whereas $\mathcal{L}_-$ is maximal HA-nonpositive.*

In connection with Theorem 1.1 note that invariant neutral subspaces (under the additional assumption that H is invertible) have been studied in [15].

The subspaces $\mathcal{L}_\pm$ of Theorem 1.1 may have additional spectral properties. To formulate these properties, it will be convenient to assume that the skew-symmetric matrix H is invertible. Let A be an H-skew-adjoint matrix. Then A is similar to $-A$, and therefore the set of eigenvalues of A is symmetric relative to both the real and the imaginary axis: if $\lambda \in \sigma(A)$, then $\pm\overline{\lambda} \in \sigma(A)$. A set of eigenvalues $\mathcal{S}$ of A will be call a *c-set* (the terminology is borrowed from [9]) if the following four conditions are fulfilled: (1) the eigenvalues in $\mathcal{S}$ all have nonzero real parts; (2) if $\lambda_0 \in \mathcal{S}$, then $\overline{\lambda_0} \in \mathcal{S}$; (3) if $\lambda_0 \in \mathcal{S}$, then $-\lambda_0 \notin \mathcal{S}$; (4) $\mathcal{S}$ is a maximal (in the sense of sets containment) set of eigenvalues of A that satisfies conditions (1), (2), and (3).

THEOREM 1.2. *Under the hypotheses of Theorem* 1.1*, assume in addition that H is invertible and that the set $\mathcal{S}_0$ of eigenvalues with nonzero real parts of the restriction $A|_{\mathcal{N}}$ is such that*

$$\lambda_0 \in \mathcal{S}_0 \quad \Longrightarrow \quad -\lambda_0 \notin \mathcal{S}_0.$$

Then for every c-set $\mathcal{S}$ such that $\mathcal{S} \supseteq \mathcal{S}_0$ there exist subspaces $\mathcal{L}_\pm$ as in Theorem 1.1 *with the additional property that $\mathcal{S}$ coincides with the set of eigenvalues with nonzero real parts of $A|_{\mathcal{L}_\pm}$.*

The cases when $\mathcal{N} = \{0\}$ and/or when $\mathcal{S}_0 = \emptyset$ are not excluded in Theorems 1.1 and 1.2.

Under the additional hypotheses that H is invertible (in Theorem 1.1) and A is invertible, these theorems were proved in [20]. The case when A is singular presents additional difficulties largely due to the fact that (assuming H is invertible) the spectrum of A has double symmetry: it is symmetric with respect to both the real axis and the imaginary axis. These two symmetries come together at the eigenvalue 0.

Theorem 1.2 can be extended to the case when H is singular, at the expense of accounting for the spectrum of $A|_{\operatorname{Ker} H}$; since the formulation of the extended result is somewhat cumbersome, we leave it out.

2. Proofs of Theorems 1.1 and 1.2. For future reference, we present a lemma.

LEMMA 2.1. *Let $Z \in \mathbb{R}^{m\times m}$ be a symmetric matrix partitioned as follows:*

$$Z = \begin{pmatrix} 0 & 0 & Q_1^T \\ 0 & Q_2 & K_1^T \\ Q_1 & K_1 & K_2 \end{pmatrix},$$

where the $p \times q$ block Q_1 is right invertible (and thus $p \le q$). Then

$$\nu_\pm(Z) + \nu_0(Z) = q + \nu_\pm(Q_2) + \nu_0(Q_2).$$

Proof. Let $Q_1^{[-1]}$ be a right inverse of Q_1, and let

$$X = -K_1^T \left(Q_1^{[-1]}\right)^T, \qquad Y = -\frac{1}{2} K_2^T \left(Q_1^{[-1]}\right)^T.$$

Then

$$\begin{pmatrix} I & 0 & 0 \\ X & I & 0 \\ Y & 0 & I \end{pmatrix} Z \begin{pmatrix} I & X^T & Y^T \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix} = \begin{pmatrix} 0 & 0 & Q_1^T \\ 0 & Q_2 & 0 \\ Q_1 & 0 & 0 \end{pmatrix}$$

and (cf. [3, Theorem 2.1])

$$\nu_\pm \begin{pmatrix} 0 & Q_1^T \\ Q_1 & 0 \end{pmatrix} + \nu_0 \begin{pmatrix} 0 & Q_1^T \\ Q_1 & 0 \end{pmatrix} = q. \qquad \square$$

Proof of Theorems 1.1 *and* 1.2. We prove these results only for HA-nonnegative subspaces. (For nonpositive subspaces the proof is analogous, or else use $-H$ in place of H.) Applying a transformation

$$A,\ H \quad \mapsto \quad S^{-1}AS,\ S^THS \tag{2.1}$$

for a suitable invertible matrix S, we may assume that H has the block form $H = \left(\begin{smallmatrix} H_1 & 0 \\ 0 & 0 \end{smallmatrix}\right)$, where H_1 is invertible. Since A is H-skew-adjoint, we obtain that A has the conformally partitioned block form $A = \left(\begin{smallmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{smallmatrix}\right)$, where A_{11} is H_1-skew-adjoint. Without loss of generality, we may assume that $\mathcal{N} \supseteq \operatorname{Ker} H$; then the proof of Theorem 1.1 is easily reduced to the situation where A and H are replaced by A_{11} and H_1, respectively; i.e., we may assume that H is invertible.

The canonical form of the pair of matrices (A, H) (with invertible H) under the transformations (2.1) (see, for example, [5, 15, 18] and in a different setup [22]) allows us to reduce the proofs to separate consideration of two cases: (1) A is invertible; (2) A is nilpotent. In the first case Theorems 1.1 and 1.2 were proved in [20, Lemma 5.2], whereas in the second case Theorem 1.2 reduces to Theorem 1.1. Thus, it remains to prove Theorem 1.1 in the case when H is invertible (and then m is necessarily even) and A is nilpotent.

We assume therefore that H is invertible and A is nilpotent, and we prove Theorem 1.1 under this assumption. Consider the subspace

$$\mathcal{N}^{[\perp]} := \{x \in \mathbb{R}^m \,|x^T H y = 0 \quad \text{for all} \quad y \in \mathcal{N}\},$$

the H-orthogonal companion of $\mathcal{N}$. As $\mathcal{N}$ is H-neutral, we have $\mathcal{N} \subseteq \mathcal{N}^{[\perp]}$. Since A is H-skew-adjoint and $\mathcal{N}$ is A-invariant, the subspace $\mathcal{N}^{[\perp]}$ is A-invariant as well. Assuming $\mathcal{N} \neq \mathcal{N}^{[\perp]}$, choose an ordered euclidean orthonormal basis

$$(y_1, \dots, y_m) \tag{2.2}$$

in $\mathbb{R}^m$ so that the first vectors in (2.2) form a basis of $\mathcal{N}$, the next vectors in (2.2) form a basis of the euclidean orthogonal complement of $\mathcal{N}$ in $\mathcal{N}^{[\perp]}$, and the remaining vectors in (2.2) form a basis of the euclidean orthogonal complement of $\mathcal{N}^{[\perp]}$ in $\mathbb{R}^m$. With respect to the basis (2.2), A has a block form

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{22} \\ 0 & A_{22} & A_{23} \\ 0 & 0 & A_{33} \end{pmatrix},$$

and the corresponding representation of H is

$$H = \left[y_i^T H y_j\right]_{i,j=1}^m = \begin{pmatrix} 0 & 0 & H_{13} \\ 0 & H_{22} & H_{23} \\ -H_{13}^T & -H_{23}^T & H_{33} \end{pmatrix}.$$

The matrix H_{22} is skew-symmetric, and A_{22} is H_{22}-skew-adjoint and nilpotent. Let $x_0 \in \operatorname{Ker} A_{22} \setminus \{0\}$. Then the subspace $\mathcal{N} + \operatorname{Span}\{x_0\}$ is clearly A-invariant and H-neutral. Now we repeat the above procedure with $\mathcal{N}$ replaced with $\mathcal{N} + \operatorname{Span}\{x_0\}$. Eventually, we reduce the proof to the case when

$$\mathcal{N} = \mathcal{N}^{[\perp]}. \tag{2.3}$$

In this case, since

$$\dim \mathcal{N} = m - \dim \mathcal{N}^{[\perp]}$$

(this equality holding because $\mathcal{N}^{[\perp]}$ coincides with the euclidean orthogonal complement to the $(\dim \mathcal{N})$-dimensional subspace $H(\mathcal{N})$), we have

$$\dim \mathcal{N} = \dim \mathcal{N}^{[\perp]} = \frac{m}{2}. \tag{2.4}$$

Thus, we assume (in addition to the assumptions made before) that (2.3) and (2.4) hold. Choosing a euclidean orthogonal basis in $\mathbb{R}^m$ such that the first half of its elements form a basis in $\mathcal{N}$, we represent A and H in the form

$$A = \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix}, \qquad B_{i,j} \in \mathbb{R}^{m/2 \times m/2}, \qquad H = \begin{pmatrix} 0 & H_1 \\ -H_1^T & H_2 \end{pmatrix}. \tag{2.5}$$

Here

$$\mathcal{N} = \operatorname{Span}\{e_1, \dots, e_{m/2}\},$$

where e_j is the jth unit coordinate vector in $\mathbb{R}^m$, $j = 1, \dots, m$, the matrix H_1 is invertible (because H is so), and H_2 is skew-symmetric. Applying a transformation (2.1) with $S = \left(\begin{smallmatrix} I & W_1 \\ 0 & W_2 \end{smallmatrix}\right)$ for suitable W_1 and W_2, we may (and do) assume that in fact

$$H = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}. \tag{2.6}$$

Then, since A is H-skew-adjoint, we have

$$A = \begin{pmatrix} B_{11} & B_{12} \\ 0 & -B_{11}^T \end{pmatrix}, \qquad B_{12} \quad \text{symmetric.} \tag{2.7}$$

Next, with A and H given by (2.6) and (2.7), we apply a transformation (2.1) with S of the form $S = \left(\begin{smallmatrix} U & 0 \\ 0 & V \end{smallmatrix}\right)$, where the invertible matrices U and V are chosen so that $U^T V = I$ and

$$U^{-1} B_{11} U = \begin{pmatrix} 0_{r \times r} & 0 \\ C_1 & C_2 \end{pmatrix}, \qquad r = \dim\left(\operatorname{Ker} B_{11}\right).$$

The matrix $[C_1 \;\; C_2]$ is clearly right invertible, the matrix H given by (2.6) being fixed under this transformation, whereas the transformed matrix A (which will be again denoted by A) is of the form

$$A = \begin{pmatrix} 0 & 0 & D_1 & D_2 \\ C_1 & C_2 & D_3 & D_4 \\ 0 & 0 & 0 & -C_1^T \\ 0 & 0 & 0 & -C_2^T \end{pmatrix}.$$

Thus,

$$HA = \begin{pmatrix} 0 & 0 & 0 & -C_1^T \\ 0 & 0 & 0 & -C_2^T \\ 0 & 0 & -D_1 & -D_2 \\ -C_1 & -C_2 & -D_3 & -D_4 \end{pmatrix}.$$

Since HA is symmetric, we have $D_1 = D_1^T$, $D_4 = D_4^T$, and $D_3 = D_2^T$. Let $\mathcal{M}_+$ be a maximal $(-D_1)$-nonnegative subspace, and let $\mathcal{L}_+ = \mathcal{N} + \mathcal{M}_+$. By Lemma 2.1,

$$\dim \mathcal{L}_+ = \nu_+(HA) + \nu_0(HA).$$

Also, $\mathcal{L}_+$ is clearly HA-nonnegative and A-invariant. This concludes the proof.

3. Riccati equations. Consider the continuous algebraic Riccati equation

$$XDX + XA + A^TX - C = 0, \tag{3.1}$$

where A, D, and C are given $n \times n$ matrices, and X is the matrix unknown. We assume throughout this section that D and C are symmetric and that D is positive semidefinite. The $2n \times 2n$ matrices

$$M = \begin{pmatrix} A & D \\ C & -A^T \end{pmatrix}, \qquad H = \begin{pmatrix} 0 & I_{n\times n} \\ -I_{n\times n} & 0 \end{pmatrix}$$

are crucial in the study of (3.1). Note that H is skew-symmetric and M is H-skew-adjoint. The *neutrality index* $\gamma(M, H)$ is defined as the maximal dimension of an M-invariant H-neutral subspace in $\mathbb{R}^{2n}$. (It was proved in [15] that all maximal (by containment) real M-invariant H-neutral subspaces have the same dimension.) This notion was introduced in [12] in the context of complex matrices that are self-adjoint in a sesquilinear inner product. The pair (A, D) is called *sign controllable* if for every $\lambda \in \mathbb{R}$ at least one of the two subspaces Ker $((\lambda I - A)^n)$ and Ker $((-\lambda I - A)^n)$ is contained in the controllable subspace

$$\mathcal{C}(A, D) := \text{Range } \left([D \;\; AD \;\; A^2D \;\; \dots \;\; A^{n-1}D]\right) \subseteq \mathbb{R}^n \tag{3.2}$$

and for every complex number $\lambda + i\mu$, where λ and μ are real and $\mu \neq 0$, at least one of the two subspaces

$$\text{Ker } \left((\lambda^2 + \mu^2)I \pm 2\lambda A + A^2\right)^n$$

is contained in $\mathcal{C}(A, D)$. The notion of sign controllability is well known in control systems, in particular in studies of algebraic Riccati equations [6, 21].

THEOREM 3.1. *Assume that the pair (A, D) is sign controllable. Then the following statements are equivalent:*

(1) *Equation* (3.1) *has a solution* $X \in \mathbb{R}^{n\times n}$.
(2) *Equation* (3.1) *has a solution* $X \in \mathbb{R}^{n\times n}$ *for which*

$$\operatorname{rank}(X - X^T) \le 2(n - \gamma(M, H)).$$

(3) *The matrix* M *has a real* n*-dimensional invariant subspace.*

Note that, for any real matrix M, statement (3) holds true if and only if either n is even, or n is odd and M has a real eigenvalue.

Under the additional hypotheses that M is invertible, Theorem 3.1 was proved in [20]. The proof of Theorem 3.1 proceeds in the same way as the proof of [20, Theorem 2.2], using Theorems 1.1 and 1.2 instead of [20, Lemma 5.2].

If $\gamma(M, H) = 0$ in Theorem 3.1, then we recover a basic result proved in [8] on the existence of symmetric solutions of algebraic Riccati equations.

The discrete algebraic Riccati equation has the form (one of several well-studied forms in the literature)

$$X = A^T XA + Q - A^T XB(R + B^* XB)^{-1} B^T XA, \tag{3.3}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $Q = Q^T \in \mathbb{R}^{n\times n}$, and $R = R^T \in \mathbb{R}^{m\times m}$ are given matrices, with the unknown $X \in \mathbb{R}^{n\times n}$. Applying a method of reduction of (3.3) to the continuous Riccati equation (see [13, 20] and [16, Chapter 12]), Theorem 3.1 yields a result concerning the existence of solutions of (3.3). We give only the necessary definitions and state the result, omitting further details (which can be easily adapted from [20] and [16, Chapter 12]).

Assume that A and R are invertible, and define the matrix

$$T = \begin{pmatrix} A + BR^{-1}B^T(A^T)^{-1}Q & -BR^{-1}B^T(A^T)^{-1} \\ -(A^T)^{-1}Q & (A^T)^{-1} \end{pmatrix}.$$

A pair of matrices (A, B), where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, is called *d-sign controllable* if for every nonzero real λ at least of the two subspaces $\operatorname{Ker}((\lambda I - A)^n)$ and $\operatorname{Ker}\left((\lambda^{-1} I - A)^n\right)$ is contained in the controllable subspace $\mathcal{C}(A, B)$ (defined in (3.2)) and for every nonzero complex number $\lambda + i\mu$, where λ and μ are real and $\mu \neq 0$, at least one of the two subspaces

$$\operatorname{Ker}\left((\lambda^2 + \mu^2)I - 2\lambda A + A^2\right)^n$$

and

$$\operatorname{Ker}\left((u^2 + v^2)I - 2uA + A^2\right)^n \quad (u + iv = (\lambda + i\mu)^{-1})$$

is contained in $\mathcal{C}(A, B)$.

THEOREM 3.2. *Let* (3.3) *be given, and assume that* A *and* R *are invertible and that the pair* (A, B) *is d-sign controllable. Further assume that there exists* $\eta \in \{1, -1\}$ *such that* η *is not an eigenvalue of* A, A^{-1}, *and* T *and that, moreover, the matrix*

$$R^{-1} - \begin{pmatrix} -R^{-1}B^T(A^T)^{-1}Q & R^{-1}B^T(A^T)^{-1} \end{pmatrix} (\eta I - T)^{-1} \begin{pmatrix} BR^{-1} \\ 0 \end{pmatrix} \tag{3.4}$$

is positive definite symmetric. Then (3.3) *admits a solution* X *such that*

$$\operatorname{rank}(X - X^T) \le 2(n - \gamma(T, J)), \qquad J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix},$$

where $\gamma(T, J)$ is the maximal dimension of a T-invariant J-neutral subspace.

Under an additional invertibility condition, Theorem 3.2 was proved in [20]. Again, specializing to the case when $\gamma(T, J) = 0$ recovers a known result on the existence of symmetric solutions of (3.3).

REFERENCES

[1] H. Abou-Kandil, G. Freiling, V. Ionescu, and G. Jank, *Matrix Riccati Equations in Control and Systems Theory*, Birkhäuser Verlag, Basel, 2003.

[2] D. Alpay, J. A. Ball, I. Gohberg, and L. Rodman, *Realization and factorization for rational matrix functions with symmetries*, in Extensions and Interpolation of Linear Operators and Matrix Functions, Oper. Theory Adv. Appl. 47, Birkhäuser, Basel, 1990, pp. 1–60.

[3] D. Alpay and H. Dym, *Structured invariant spaces of vector valued rational functions, Hermitian matrices, and a generalization of the Iohvidov laws*, Linear Algebra Appl., 137/138 (1990), pp. 137–181.

[4] L. Barkwell, P. Lancaster, and A. S. Markus, *Gyroscopically stabilized systems: A class of quadratic eigenvalue problems with real spectrum*, Canad. J. Math., 44 (1992), pp. 42–53.

[5] D. Z. Djokovic, J. Patera, P. Winternitz, and H. Zassenhaus, *Normal forms of elements of classical real and complex Lie and Jordan algebras*, J. Math. Phys., 24 (1983), pp. 1363–1374.

[6] L. E. Faibusovich, *Algebraic Riccati equation and symplectic algebra*, Internat. J. Control, 43 (1986), pp. 781–792.

[7] H. Fassbender, D. S. Mackey, N. Mackey, and H. Xu, *Hamiltonian square roots of skew-Hamiltonian matrices*, Linear Algebra Appl., 287 (1999), pp. 125–159.

[8] P. A. Fuhrmann, *On Hamiltonian rational transfer functions*, Linear Algebra Appl., 63 (1984), pp. 1–93.

[9] I. Gohberg, P. Lancaster, and L. Rodman, *Matrices and Indefinite Scalar Products*, Oper. Theory Adv. Appl. 8, Birkhäuser Verlag, Basel, 1983.

[10] V. Ionescu, C. Oară, and M. Weiss, *Generalized Riccati Theory and Robust Control*, John Wiley and Sons, Chichester, 1999.

[11] P. Lancaster, A. S. Markus, and F. Zhou, *A wider class of stable gyroscopic systems*, Linear Algebra Appl., 370 (2003), pp. 257–267.

[12] P. Lancaster, A. S. Markus, and Q. Ye, *Low rank perturbations of strongly definitizable transformations and matrix polynomials*, Linear Algebra Appl., 197/198 (1994), pp. 3–29.

[13] P. Lancaster, A. C. M. Ran, and L. Rodman, *Hermitian solutions of the discrete algebraic Riccati equation*, Internat. J. Control, 44 (1986), pp. 777–802.

[14] P. Lancaster and L. Rodman, *Minimal symmetric factorizations of symmetric real and complex rational matrix functions*, Linear Algebra Appl., 220 (1995), pp. 249–282.

[15] P. Lancaster and L. Rodman, *Invariant neutral subspaces for symmetric and skew real matrix pairs*, Canad. J. Math., 46 (1994), pp. 602–618.

[16] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, Oxford University Press, New York, 1995.

[17] 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.

[18] A. C. M. Ran and L. Rodman, *Stability of invariant Lagrangian subspaces*, I., in Topics in Operator Theory, Oper. Theory Adv. Appl. 32, Birkhäuser, Basel, 1988, pp. 181–218.

[19] A. C. M. Ran and L. Rodman, *Stable invariant Lagrangian subspaces: Factorization of symmetric rational matrix functions and other applications*, Linear Algebra Appl., 137/138 (1990), pp. 575–620.

[20] L. Rodman, *Non-Hermitian solutions of algebraic Riccati equations*, Canad. J. Math., 49 (1997), pp. 840–854.

[21] C. Scherer, *The solution set of the algebraic Riccati equation and the algebraic Riccati inequality*, Linear Algebra Appl., 153 (1991), pp. 99–122.

[22] R. C. Thompson, *Pencils of complex and real symmetric and skew matrices*, Linear Algebra Appl., 147 (1991), pp. 323–371.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 4, pp. 908–917

EXCLUSION AND INCLUSION INTERVALS FOR THE REAL EIGENVALUES OF POSITIVE MATRICES*

J. M. PEÑA†

Abstract. Given a real matrix, we analyze an open interval, called a row exclusion interval, such that the real eigenvalues do not belong to it. We characterize when the row exclusion interval is nonempty. In addition to the exclusion interval, inclusion intervals for the real eigenvalues, alternative to those provided by the Gerschgorin disks, are also considered for matrices whose off-diagonal entries present a restricted dispersion. The results are applied to obtain a sharp upper bound for the real eigenvalues different from 1 of a positive stochastic matrix and a sufficient condition for the stability of a negative matrix, among other applications.

Key words. Gerschgorin circles, positive matrices, real eigenvalue localization, stochastic matrix, exclusion interval

AMS subject classifications. 65F15, 15A18

DOI. 10.1137/04061074X

1. Introduction. Several inclusion regions in the complex plane for the eigenvalues of a matrix have been considered: Gerschgorin disks (see [14]), Brauer ovals of Cassini (see [1] and [15]), Brualdi lemniscata sets (see [3]), or the minimal Gerschgorin set (see [12]). These sets have recently been compared in [13], and in [4] other inclusion regions appear. In order to localize the real parts of the eigenvalues of a real matrix, alternative methods to Gerschgorin disks and Brauer ovals of Cassini have been presented in [10] and [11], respectively. A key tool for these alternative methods has been the use of a class of real matrices with positive determinant, called B-matrices.

We consider in section 2 another class of nonsingular real matrices, called C-matrices. In Proposition 2.6, we use C-matrices in order to obtain an open interval (called the row exclusion interval) associated with a real matrix and such that no real eigenvalue belongs to it. In Example 2.1 we prove that, in contrast to the results of [10], the row exclusion interval cannot be applied to the localization of the real parts of the eigenvalues because these real parts can belong to it.

In Proposition 3.1, we characterize when the row exclusion interval is nonempty. In the case of stochastic matrices, the row exclusion interval depends on the least off-diagonal element of the matrix. This phenomenon also happened in the context of bounding the Perron root of a positive matrix (see, for instance, section 2.1 of [9]). In section 3, we see that the class of matrices with a nonempty row exclusion interval contains the class of matrices which are multiples of a stochastic matrix, and we also apply the row exclusion interval in order to provide an upper bound of the real eigenvalues different from 1 of a stochastic matrix in terms of the least off-diagonal entry of the matrix. Example 3.1 shows that this bound cannot be improved.

In section 4 we consider matrices whose off-diagonal entries present a restricted

*Received by the editors June 29, 2004; accepted for publication by R. Bhatia August 30, 2004; published electronically May 6, 2005. This research was partially supported by the Spanish research grant BFM2003-03510.

http://www.siam.org/journals/simax/26-4/61074.html

†Departamento de Matemática Aplicada, Universidad de Zaragoza, 50009 Zaragoza, Spain (jmpena@unizar.es).

dispersion. In particular, the results can be applied to matrices A satisfying

$$sJ \leq A \leq 2sJ,$$

where $s > 0$ and J is the matrix of ones. In Theorem 4.1 we show that, for these matrices, the row exclusion interval is nonempty and the inclusion intervals for the real eigenvalues obtained in [10], called $\bar{B}$-intervals, provide sharper information than the real intervals provided by Gerschgorin circles, and we also obtain a lower bound for the real parts of all the eigenvalues. In fact, analogously to the property that Gerschgorin disks provide sharper information when the matrix resembles a diagonally dominant matrix, the (inclusion) $\bar{B}$-intervals and the exclusion interval provide sharper information when the off-diagonal entries decrease their dispersion. In Corollary 4.2, we give a sufficient condition for the stability of a negative matrix. Finally, we derive some applications to Toeplitz matrices.

2. *C*-matrices and the exclusion interval. Let us start by introducing a class of nonsingular matrices.

DEFINITION 2.1. *We say that a square real matrix $A = (a_{ik})_{1\leq i,k\leq n}$ with positive row sums is a C-matrix if all its off-diagonal elements are bounded below by the corresponding row means, i.e., for all $i = 1, \ldots, n$*

$$\sum_{k=1}^{n} a_{ik} > 0 \quad \textit{and} \quad \frac{1}{n}\left(\sum_{k=1}^{n} a_{ik}\right) < a_{ij} \ \ \forall\, j \neq i.$$

Remark 2.1. From the previous definition we can deduce that all the off-diagonal elements of a C-matrix are positive and the diagonal elements of a C-matrix satisfy for all $i = 1, \ldots, n$

$$a_{ii} < \min\{a_{ij} \mid j \neq i\},$$

and therefore each row mean of a C-matrix is bounded below by the maximal between 0 and the diagonal element and bounded above by any off-diagonal element of the row.

Although we could derive the nonsingularity of a C-matrix from an adequate application of Theorem 4.4 of [5], for the sake of completeness we provide a direct proof of this fact.

LEMMA 2.2. *If A is an $n \times n$ C-matrix, then $(-1)^{n-1} \det A > 0$.*

Proof. Let $e := (1, \ldots, 1)^T$, and let $m := \frac{1}{n}Ae$ be the vector of row means. If $m = (m_1, \ldots, m_n)^T$, let $d > 0$ be such that

$$1 + 2d = \min_{j=1,\ldots,n} \left(\min_{k\neq j} \frac{a_{kj}}{m_j}\right).$$

Then, for all $i = 1, \ldots, n$ and $j \neq i$,

$$a_{ij} \geq (1 + 2d)m_i > (1 + d)m_i > 0. \tag{2.1}$$

The identity matrix will be denoted by I. If we define the matrices $P := \frac{1+d}{n}ee^T - I$ and $M := AP$, then $m_{ij} = (1+d)m_i - a_{ij}$ for all i, j. By construction, M is a Z-matrix (i.e., a matrix whose off-diagonal elements are nonpositive) because, by (2.1), its off-diagonal elements are negative. In addition, the row sums of M are positive because $Me = (1 + d)nm - nm = dnm$ is a vector with positive components. This means

that M is a matrix strictly diagonally dominant by rows and has positive diagonal entries. Then it is well known that $\det M > 0$ (cf. [5] or, for a direct proof, use the Gerschgorin circles to see that M has its eigenvalues with positive real part). Since the eigenvalues of P are d and -1 (with multiplicity $n-1$), we get $\det P = (-1)^{n-1}d$, and the result follows from taking determinants in $M = AP$. □

Let $A = (a_{ik})_{1\leq i,k\leq n}$ be a real matrix. From now on, we shall use the following notation: for each $i = 1, \ldots, n$

$$s_i^+ := \max\{0, \min\{a_{ij} \mid j \neq i\}\}, \quad s_i^- := \min\{0, \max\{a_{ij} \mid j \neq i\}\}. \tag{2.2}$$

Let us remark that

$$s_i^+ s_i^- = 0, \quad i = 1, \ldots, n. \tag{2.3}$$

The next result provides a characterization of C-matrices which can be derived from Definition 2.1.

PROPOSITION 2.3. *Let $A = (a_{ik})_{1\leq i,k\leq n}$ be a real matrix, and, for each $i = 1, \ldots, n$, let s_i^+ be the number given in (2.2). Then A is a C-matrix if and only if for all $i \in \{1, \ldots, n\}$*

$$0 < \sum_{k=1}^{n} a_{ik} < n s_i^+.$$

We now introduce a class of nonsingular matrices closely related to C-matrices.

DEFINITION 2.4. *We say that a real matrix is a $\bar{C}$-matrix if it is of the form DA, where D is a diagonal matrix whose diagonal elements belong to the set $\{1, -1\}$ and A is a C-matrix.*

Remark 2.2. If either A or A^T is a $\bar{C}$-matrix, then it is a nonsingular matrix because, by Lemma 2.2, C-matrices are nonsingular.

Remark 2.3. If A is a $\bar{C}$-matrix, then all its off-diagonal elements are nonzero. In addition, for each row i of A the off-diagonal elements of row i agree in sign.

The following characterization of $\bar{C}$-matrices is a consequence of Proposition 2.3 and Definition 2.4.

PROPOSITION 2.5. *Let $A = (a_{ik})_{1\leq i,k\leq n}$ be a real matrix, and, for each $i = 1, \ldots, n$, let s_i^+, s_i^- be as in (2.2). Then A is a $\bar{C}$-matrix if and only if for each $i \in \{1, \ldots, n\}$ either*

$$0 < \sum_{k=1}^{n} a_{ik} < n s_i^+$$

or

$$0 > \sum_{k=1}^{n} a_{ik} > n s_i^-.$$

The following result provides information on the localization of the real eigenvalues of a real matrix.

PROPOSITION 2.6. *Let $A = (a_{ik})_{1\leq i,k\leq n}$ be a real matrix, let s_i^+, s_i^- be as in (2.2), and let λ be a real eigenvalue of A. Then*

$$\lambda \notin E := \left(\max_{i=1,\ldots,n} \left\{ \sum_{j=1}^{n} a_{ij} - n s_i^+ \right\}, \min_{i=1,\ldots,n} \left\{ \sum_{j=1}^{n} a_{ij} - n s_i^- \right\} \right). \tag{2.4}$$

Proof. Let $e := (1, \dots, 1)^T$ and $r := Ae$. Therefore

$$r_i := \sum_{j=1}^{n} a_{ij} \tag{2.5}$$

for all $i = 1, \dots, n$. From the definition of E, we see that, for each i and for any $t \in E$,

$$(r_i - t) - ns_i^+ < 0 < (r_i - t) - ns_i^-.$$

If $t = r_k$ for some k, then $-ns_k^+ < 0 < -ns_k^-$, which contradicts (2.3). Thus no element in E can equal any row sum. Now consider $A - tI$. For each i, either

$$ns_i^+ > r_i - t > 0$$

or

$$ns_i^- < r_i - t < 0.$$

By Proposition 2.5, $A - tI$ is a $\bar{C}$-matrix and is thus nonsingular. ∎

The interval E of the previous result will play a key role in this paper.

DEFINITION 2.7. *The interval E appearing in* (2.4) *will be called the row exclusion interval.*

By using A^T instead of A and applying Remark 2.2, we could prove a result similar to Proposition 2.6 but changing the role of the rows by columns, and we also could define a column exclusion interval. The remaining results of this paper could also be adapted into a version "by columns."

COROLLARY 2.8. *Let $A = (a_{ik})_{1 \le i,k \le n}$ be a real matrix. If A has a row i with two off-diagonal elements of different signs or with some null entries, then the row exclusion interval is empty.*

Proof. Clearly, $s_i^- = 0 = s_i^+$, and so the second endpoint of the (open) row exclusion interval is bounded above by $\sum_{j=1}^{n} a_{ij}$, which in turn is less than or equal to the first endpoint. ∎

As the previous result shows, Proposition 2.6 provides information on the localization of the real eigenvalues when the matrix has the off-diagonal elements of each row of the same sign. This happens, for instance, with the Z-matrices or the matrices opposite Z-matrices.

A P-matrix is a matrix such that all its leading principal minors are positive. In [10], a class of P-matrices (called B-matrices) was considered in order to obtain localization results on the real eigenvalues of a real matrix. In that paper, the results could be extended (using a property of P-matrices) in order to localize the real parts of the eigenvalues of a real matrix. So, it is natural to ask whether Proposition 2.6 is also valid replacing "real eigenvalues" by "real parts of the eigenvalues." The following example shows that this extension is not possible and also shows the deep differences between both situations. (A result similar to Corollary 4.2 of [10] does not hold here: given a complex matrix A whose off-diagonal entries are real, A can be singular, although the real parts $\mathrm{Re}(A)$ and $\mathrm{Re}(A^T)$ are $\bar{C}$-matrices.)

Example 2.1. The matrix

$$B = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

has $s_1^+ = 0 = s_2^-$, $s_1^- = -1$, $s_2^+ = 1$, and so the row exclusion interval is $E = (-1, 1)$, which contains the real parts of its eigenvalues $\pm i$. In fact, the matrix

$$A = \begin{pmatrix} i & -1 \\ 1 & i \end{pmatrix}$$

is singular, although $\mathrm{Re}(A) = B$ and $\mathrm{Re}(A^T) = B^T$ are $\bar{C}$-matrices. In contrast to B, the symmetric matrix

$$C = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

also has the row exclusion interval $E = (-1, 1)$, and its eigenvalues are the endpoints of the intervals provided by the Gerschgorin circles.

3. Bounds for the real eigenvalues of a positive matrix. Given a nonnegative matrix $A = (a_{ik})_{1 \le i,k \le n}$, let us recall that the number $r_i = \sum_{j=1}^n a_{ij}$ defined in (2.5) is the right endpoint of the real interval provided by the corresponding row Gerschgorin circle for each $i = 1, \ldots, n$. Let us also define

$$\rho := \min\{r_i |\, i = 1, \ldots, n\}, \quad R := \max\{r_i |\, i = 1, \ldots, n\}. \tag{3.1}$$

The following proposition characterizes the matrices whose row exclusion interval is nonempty.

PROPOSITION 3.1. *Let $A = (a_{ik})_{1 \le i,k \le n}$ be a real matrix for each $i = 1, \ldots, n$, let s_i^+ and r_i be the numbers given by* (2.2) *and* (2.5)*, respectively, and let R and ρ be the numbers given by* (3.1)*. Then the row exclusion interval is nonempty if and only if either its off-diagonal entries are positive and*

$$r_i - n s_i^+ < \rho, \quad i = 1, \ldots, n, \tag{3.2}$$

or its off-diagonal entries are negative and

$$R < r_i - n s_i^-, \quad i = 1, \ldots, n.$$

Proof. By Corollary 2.8, a matrix with a nonempty row exclusion interval must have either all its off-diagonal elements positive or all its off-diagonal elements negative. If A has positive off-diagonal entries, $s_i^- = 0$ for all $i = 1, \ldots, n$. By Proposition 2.6, the row exclusion interval is

$$E = (\max\{r_i - n s_i^+ |\, i = 1, \ldots, n\}, \rho), \tag{3.3}$$

and so E is nonempty if and only if (3.2) holds. The case when A has negative off-diagonal entries can be proved analogously. ∎

Let us recall that a nonnegative matrix is called *row stochastic* (or simply *stochastic*) if all its row sums are 1. The following result provides a new upper bound of the real eigenvalues different from 1 of a stochastic matrix in terms of the least off-diagonal element.

PROPOSITION 3.2. *Let $A = (a_{ik})_{1 \le i,k \le n}$ be a stochastic matrix, and let s^+ and w be the least off-diagonal and diagonal entries of A, respectively. If λ is a real eigenvalue of A, then either $\lambda = 1$ (with algebraic multiplicity 1 if $s^+ > 0$) or $2w - 1 \le \lambda \le 1 - n s^+$. If, in addition, A is positive, then the row exclusion interval is nonempty.*

Proof. If

$$w = \min\{a_{ii} \mid i = 1, \dots, n\}, \tag{3.4}$$

then it is known (see [6] or Theorem 1.4 of Chapter 6 of [9] or use the Gerschgorin circles) that $|\lambda - w| \leq 1 - w$. It is well known that the eigenvalues of a stochastic matrix A have absolute value less than or equal to 1 and that 1 is an eigenvalue of A. Since $0 \leq w \leq 1$ and $\lambda \leq 1$, we obtain $2w - 1 \leq \lambda$.

The bound $\lambda \leq 1 - ns^+$ clearly holds if $s^+ = 0$ because then $1 - ns^+ = 1$. Let us now assume that $s^+ > 0$ (and so that A is positive). Then A is irreducible, and, by Theorem 4.3 of Chapter 1 of [9], the eigenvalue 1 has algebraic multiplicity 1. Observe that the number

$$s^+ = \min\{s_i^+ \mid i = 1, \dots, n\}, \tag{3.5}$$

where the numbers s_i^+ are given in (2.2). By formula (3.3), the row exclusion interval is $E = (1 - ns^+, 1)$. So, the real eigenvalues of A different from 1, which are less than 1, are in fact less than or equal to $1 - ns^+$. Finally, since $s^+ > 0$, E is nonempty. ∎

By Proposition 3.2, the class of positive matrices with a nonempty row exclusion interval contains the class of positive stochastic matrices. In fact, taking into account (3.3), any nontrivial multiple of a stochastic matrix has a nonempty row exclusion interval E.

Remark 3.1. Let us observe that the length of the row exclusion interval (2.4) of a real matrix is invariant under a scalar diagonal translation; that is, it coincides for a matrix A and for a matrix $A + D$, where $D = \operatorname{diag}\{d, \dots, d\}$. Besides, as shown in the proof of the previous proposition, the row exclusion interval of a stochastic matrix (which is given by (3.3)) depends on its least off-diagonal element.

The next example shows that the bound of Proposition 3.2 cannot be improved.

Example 3.1. Any $n \times n$ stochastic matrix whose off-diagonal entries coincide has the form

$$M = \begin{pmatrix} z & y & \cdots & \cdots & y \\ y & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & y \\ y & \cdots & \cdots & y & z \end{pmatrix}, \quad y \geq 0, \tag{3.6}$$

with $z \geq 0$ and $z + (n-1)y = 1$. Its eigenvalues are 1 and $z - y$ (with multiplicity $n - 1$). Then, since $s^+ = y$, we have $1 - ns^+ = z + (n-1)y - ny = z - y$, and so the bound of Proposition 3.2 is sharp.

A theorem due to Frobenius (see [7] or Theorem 1.1 of Chapter 2 of [9]) shows that the maximal eigenvalue of a nonnegative matrix belongs to the interval $[\rho, R]$. On the other hand, Brauer proved (see [2] or Theorem 1.5 of Chapter 2 of [9]) that the maximal eigenvalue of a positive matrix belongs to the interval

$$J = \left[\rho + \eta(h-1),\ R - \left(1 - \frac{1}{g}\right)\eta\right], \tag{3.7}$$

where $\eta := \min\{w, s^+\}$ and w, s^+ are given by (3.4) and (3.5) (i.e., η is the minimal element of A) and

$$g = \frac{R - 2\eta + \sqrt{R^2 - 4\eta(R-\rho)}}{2(\rho - \eta)}, \quad h = \frac{-\rho + 2\eta + \sqrt{\rho^2 + 4\eta(R-\rho)}}{2\eta}.$$

The following result shows that the condition (3.2), together with some additional hypotheses, implies that the interval J of (3.7) contains a unique real eigenvalue and provides an upper bound for the remaining $n-1$ real eigenvalues of A. First, we need some auxiliary notation.

Given a matrix $B = (b_{ik})_{1\leq i,k\leq n}$, let us define the family of matrices

$$B_t := S^+ + t(B - S^+), \quad t \in [0,1], \tag{3.8}$$

where

$$S^+ = \begin{pmatrix} s^+ & \cdots & s^+ \\ \vdots & & \vdots \\ s^+ & \cdots & s^+ \end{pmatrix} \tag{3.9}$$

and s^+ is given by (3.2) (i.e., $s^+ = \min\{s_i^+ \,|\, i = 1,\ldots,n\}$, where the numbers s_i^+ come from (2.2), using $B = (b_{ik})_{1\leq i,k\leq n}$ instead of $A = (a_{ik})_{1\leq i,k\leq n}$).

THEOREM 3.3. *Let $A = (a_{ik})_{1\leq i,k\leq n}$ be a positive matrix for each $i = 1,\ldots,n$, let s_i^+ and r_i be the numbers given by* (2.2) *and* (2.5)*, respectively, and let ρ be the number given by* (3.1)*. If* (3.2) *holds and, in addition, $A \in \mathcal{C}$, where $\mathcal{C}$ is a class of real matrices such that if $B \in \mathcal{C}$ all eigenvalues of B are real and all matrices of the form* (3.8) *belong to $\mathcal{C}$, then $n-1$ eigenvalues of A are less than or equal to $\max\{r_i - ns_i^+ \,|\, i = 1,\ldots,n\}$, and there exists a unique eigenvalue of A in the interval J of* (3.7)*.*

Proof. For each $i = 1,\ldots,n$ and for every $t \in [0,1]$, let $s_{i,t}^+$ and $r_{i,t}$ be the numbers given by (2.2) and (2.5) (replacing A by A_t), respectively, and let ρ_t be the number given by (3.1) (replacing r_i by $r_{i,t}$). Observe that $s_{i,t}^+ = ts_i^+ + (1-t)s^+$, $r_{i,t} = tr_i + (1-t)ns^+ \geq tr_i$. So, given $h \in \{1,\ldots,n\}$ such that $\rho_t = r_{h,t}$, $\rho_t \geq tr_h \geq t\rho$. Then, by (3.2), we deduce that, for all $t \in (0,1]$ and for all $i = 1,\ldots,n$,

$$r_{i,t} - ns_{i,t}^+ = t(r_i - ns_i^+) < t\rho \leq \rho_t.$$

So, by Proposition 3.1, the row exclusion interval E_t of each positive matrix A_t, $t \in (0,1]$, is nonempty.

On the other hand, by formula (3.3) (using $r_{i,0} = ns^+$, $s_{i,0}^+ = s^+$ for all $i = 1,\ldots,n$, and ρ_0 instead of r_i, s_i^+ and ρ, respectively), the $n \times n$ matrix $A_0 = S^+$ has the row exclusion interval $E = (0,\rho_0)$, where $\rho_0 = ns^+$ and s^+ is given by (3.2). The eigenvalues of A_0 are 0 (with multiplicity $n-1$) and $ns^+ (= \rho_0)$.

Since all the eigenvalues of the matrices A_t ($t \in [0,1]$) are real and the matrix A_0 has precisely one eigenvalue greater than or equal to ρ_0, we derive in this case from the fact that the row exclusion intervals of the matrices A_t are nonempty for all $t \in [0,1]$ and the continuity of the eigenvalues as functions of the elements of the matrix that there exists precisely one real eigenvalue greater than or equal to ρ_t. In particular, there exists a unique eigenvalue greater than or equal to $\rho = \rho_1$, and, by Theorem 1.5 of Chapter 2 of [9], there exists a unique eigenvalue in the interval J of (3.7). By Proposition 2.6, there are $n-1$ eigenvalues of A less than or equal to $\max\{r_i - ns_i^+ \,|\, i = 1,\ldots,n\}$. ∎

Observe that the previous result can be applied to the class of symmetric matrices.

In order to illustrate an application of Theorem 3.3, let us assume that a positive matrix A is positive definite symmetric and such that $\max\{r_i - ns_i^+ \,|\, i = 1,\ldots,n\} < \rho + \eta(h-1)$. Then the quotient between the second largest eigenvalue λ_2 and the

largest eigenvalue λ_1 of A satisfies

$$\frac{\lambda_2}{\lambda_1} \leq \frac{\max\{r_i - ns_i^+ \,|\, i = 1, \ldots, n\}}{\rho + \eta(h-1)},$$

and it is well known that this quotient provides information on the speed of convergence of the power method.

4. Inclusion and exclusion intervals for the real eigenvalues of matrices whose off-diagonal entries have restricted dispersion. Let us start this section by introducing a class of matrices to which we shall apply results on the localization of eigenvalues. Let A be a matrix such that all its off-diagonal elements are positive and satisfy

$$s^+ \leq a_{ij} < 2s^+, \quad i \neq j, \tag{4.1}$$

where s^+ is its least off-diagonal element (see (3.5)).

The following result shows for matrices satisfying (4.1) that the inclusion intervals for the real parts of its eigenvalues, called $\bar{B}$-intervals in [10, 11], are contained in the real intervals provided by the Gerschgorin circles. Let us recall the following notation introduced in [10]: given a real matrix $A = (a_{ik})_{1 \leq i,k \leq n}$, for each $i = 1, \ldots, n$

$$r_i^+ := \max\{0, a_{ij} \mid j \neq i\}, \quad r_i^- := \min\{0, a_{ij} \mid j \neq i\}. \tag{4.2}$$

For each row $i = 1, \ldots, n$, the corresponding *row $\bar{B}$-interval* is given by the interval

$$\left[a_{ii} - r_i^+ - \sum_{k \neq i} |r_i^+ - a_{ik}|,\ a_{ii} - r_i^- + \sum_{k \neq i} |r_i^- - a_{ik}| \right]. \tag{4.3}$$

Theorem 3.5(i) of [10] proves that the real eigenvalues of a real matrix A belong to the union of the row $\bar{B}$-intervals. Analogously, the column $\bar{B}$-intervals can be defined, and Theorem 4.3(i) of [10] proves that all the real parts of the eigenvalues of A belong to the union of the row and column $\bar{B}$-intervals.

THEOREM 4.1. *Let $A = (a_{ik})_{1 \leq i,k \leq n}$ be a matrix satisfying* (4.1)*, let w and s^+ be the numbers given by* (3.4) *and* (3.5)*, respectively, and let R be the number given by* (3.1)*. Then the following properties hold:*

(i) *For each row $i = 1, \ldots, n$, the corresponding row $\bar{B}$-interval is contained in the real interval provided by the corresponding Gerschgorin circle.*

(ii) *The interval $[w - ns^+, R]$ contains all the row $\bar{B}$-intervals. Besides, all the real parts of the eigenvalues of A are bounded below by $w - ns^+$.*

(iii) *If, in addition, the diagonal elements satisfy*

$$\max_{i=1,\ldots,n} a_{ii} - \min_{i=1,\ldots,n} a_{ii} < s^+,$$

then the row exclusion interval contains the nonempty interval $E' = (R - ns^+, \rho)$.

Proof. (i) Given a positive matrix $A = (a_{ik})_{1 \leq i,k \leq n}$, the right endpoints of the row $\bar{B}$-intervals (4.3) and the right endpoints of the real intervals provided by Gerschgorin row-regions coincide. Since A is positive, for each $i = 1, \ldots, n$ there exists $j \neq i$ such

that $r_i^+ = a_{ij}$. The left endpoints of the real intervals provided by Gerschgorin row-regions are given by

$$a_{ii} - a_{ij} - \sum_{k \neq i,j} a_{ik}, \quad i = 1, \ldots, n, \tag{4.4}$$

and the left endpoints of the row $\bar{B}$-intervals (4.3) can be written as

$$a_{ii} - nr_i^+ + \sum_{k \neq i} a_{ik} = a_{ii} - a_{ij} - (n-2)r_i^+ + \sum_{k \neq i,j} a_{ik}, \quad i = 1, \ldots, n. \tag{4.5}$$

Since A satisfies (4.1), $r_i^+ - a_{ik} < s^+ \leq a_{ik}$ $(k \neq i, j)$, and then we can deduce that each number of (4.5) is greater than the corresponding number of (4.4) and (i) holds.

(ii) Again, let $j \neq i$ be such that $r_i^+ = a_{ij}$. The left endpoints of the row $\bar{B}$-intervals (4.3) can be written as in (4.5). Since A satisfies (4.1), we derive $r_i^+ - a_{ik} < s^+$ when $k \neq i$, and, taking into account that $a_{ij} < 2s^+$, the elements of (4.5) are greater than the corresponding numbers

$$a_{ii} - ns^+, \quad i = 1, \ldots, n, \tag{4.6}$$

which, in turn, are bounded below by $w - ns^+$. Since the right endpoints of the row $\bar{B}$-intervals (4.3) and the right endpoints of the real intervals provided by the Gerschgorin row-regions coincide (because A is positive), they are bounded above by R, and so the interval $[w - ns^+, R]$ contains all the row $\bar{B}$-intervals. Applying to A^T the reasoning of the beginning of this paragraph we also conclude that the left endpoints of its row $\bar{B}$-intervals (which are the left endpoints of the column $\bar{B}$-intervals of A) are bounded below by $w - ns^+$. Then, since, by Theorem 4.3(i) of [10], all the real parts of the eigenvalues of A belong to the union of the row and column $\bar{B}$-intervals, (ii) follows.

(iii) Since A has positive off-diagonal entries, we deduce from formula (3.3) that the row exclusion interval of A is $E = (\max\{r_i - ns_i^+ \,|\, i = 1, \ldots, n\}, \rho)$, which clearly contains E'. Let us now prove that $R - \rho < ns^+$. Without loss of generality we may assume that $w = s^+$. By the hypotheses satisfied by all entries of A, we can deduce that $ns^+ \leq R, \rho < 2ns^+$. Hence $0 \leq R - ns^+ < ns^+ \leq \rho$. Thus $R - \rho < ns^+$, and so E' is nonempty. □

From Theorem 4.1(ii), we can derive the following sufficient condition for a positive stable matrix.

COROLLARY 4.2. *Let $A = (a_{ik})_{1 \leq i,k \leq n}$ be a positive matrix satisfying* (4.1)*, and let w and s^+ be the numbers given by* (3.4) *and* (3.5)*, respectively. If $w > ns^+$, then the real parts of all eigenvalues of A are positive.*

Given a negative matrix, we can apply the previous corollary to $-A$, and then the following sufficient condition for stability holds.

COROLLARY 4.3. *Let $A = (a_{ik})_{1 \leq i,k \leq n}$ be a negative matrix, and let v and s^- be the maximal diagonal and off-diagonal entries of A, respectively. If, for all $i \neq j$, $2s^- < a_{ij} \leq s^-$ and, in addition, $v < ns^-$, then the real parts of all eigenvalues of A are negative.*

Any $n \times n$ matrix of the form (3.6) shows that the combination of the row $\bar{B}$-intervals and the row exclusion interval can provide sharp information on the eigenvalues. In this case, the numbers given by (4.2) are $r_i^+ = y$, $r_i^- = 0$, and so the row $\bar{B}$-intervals are $[z - y, z + (n-1)y]$. Since the numbers given by (2.2), (2.5), and (3.1) are $s_i^+ = y$ and $\rho = z + (n-1)y = R = r_i$ for all $i = 1, \ldots, n$, the row exclusion

interval is, by (3.3), $(z-y, z+(n-1)y)$. Hence we already obtain that the eigenvalues of A can only be the numbers $z-y$ and $z+(n-1)y$. On the other hand, since $\rho = R$ and $h = 1 = g$, the interval J of (3.7) is in fact the point ρ. So, by Theorem 4.1(ii), $z+(n-1)y$ is an eigenvalue with multiplicity 1, and $z-y$ is an eigenvalue with multiplicity $n-1$.

The previous matrix M is Toeplitz. Let us recall that matrices whose entries are constant along each diagonal arise in many applications and are called Toeplitz matrices (see [8]): a matrix $A = (a_{ij})_{1\le i,j\le n}$ is *Toeplitz* if there exist real numbers

$$r_{-n+1}, \ldots, r_{-1}, r_0, r_1, \ldots, r_{n-1}$$

such that $a_{ij} = r_{j-i}$ for all i, j. If a positive (resp., negative) Toeplitz matrix satisfies $\max\{r_i - r_j |\, i, j \neq 0\} < \min\{r_k |\, k \neq 0\}$ (resp., $\min\{r_i - r_j |\, i, j \neq 0\} > \max\{r_k |\, k \neq 0\}$) and $r_0 > n \min\{r_k |\, k \neq 0\}$ (resp., $r_0 < n \max\{r_k |\, k \neq 0\}$), then, by Corollary 4.2 (resp., Corollary 4.3), A is positive stable (resp., is stable).

REFERENCES

[1] A. Brauer, *Limits for the characteristic roots of a matrix* II, Duke Math. J., 14 (1947), pp. 21–26.

[2] A. Brauer, *The theorems of Ledermann and Ostrowski on positive matrices*, Duke Math. J., 24 (1957), pp. 265–274.

[3] R. Brualdi, *Matrices, eigenvalues and directed graphs*, Linear and Multilinear Algebra, 11 (1982), pp. 143–165.

[4] R. A. Brualdi and S. Mellendorf, *Regions in the complex plane containing the eigenvalues of a matrix*, Amer. Math. Monthly, 101 (1994), pp. 975–985.

[5] J. M. Carnicer, T. N. T. Goodman, and J. M. Peña, *Linear conditions for positive determinants*, Linear Algebra Appl., 292 (1999), pp. 39–59.

[6] M. Fréchet, *Comportement asymptotique de solutions d'un systeme d'équations linéaires et homogenes aux différences finies du primier ordre a coefficients constants*, Publ. Fac. Sci. Univ. Mayaryk, 178 (1933), pp. 1–24.

[7] G. Frobenius, *Über Matrizen aus nich negativen Elementen*, S.-B. K. Preuss. Akad. Wiss. Berlin, 1912, pp. 456–477.

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The John Hopkins University Press, London, 1996.

[9] H. Minc, *Nonnegative Matrices*, Wiley-Interscience, New York, 1988.

[10] J. M. Peña, *A class of P-matrices with applications to the localization of the eigenvalues of a real matrix*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1027–1037.

[11] J. M. Peña, *On an alternative to Gerschgorin circles and ovals of Cassini*, Numer. Math., 95 (2003), pp. 337–345.

[12] R. S. Varga, *Minimal Gerschgorin sets*, Pacific J. Math., 15 (1965), pp. 719–729.

[13] R. S. Varga, *Geršcgorin-type eigenvalue inclusion theorems and their sharpness*, Electron. Trans. Numer. Anal., 12 (2001), pp. 113–133.

[14] R. S. Varga, *Geršcgorin and His Circles*, Springer-Verlag, Berlin, 2004.

[15] R. S. Varga and A. Krautstengl, *On Geršgorin-type problems and ovals of Cassini*, Electron. Trans. Numer. Anal., 8 (1999), pp. 15–20.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 4, pp. 918–946

A RANK-REVEALING METHOD WITH UPDATING, DOWNDATING, AND APPLICATIONS*

T. Y. LI† AND ZHONGGANG ZENG‡

Abstract. A new rank-revealing method is proposed. For a given matrix and a threshold for near-zero singular values, by employing a globally convergent iterative scheme as well as a deflation technique the method calculates approximate singular values below the threshold one by one and returns the approximate rank of the matrix along with an orthonormal basis for the approximate null space. When a row or column is inserted or deleted, algorithms for updating/downdating the approximate rank and null space are straightforward, stable, and efficient. Numerical results exhibiting the advantages of our code over existing packages based on two-sided orthogonal rank-revealing decompositions are presented. Also presented are applications of the new algorithm in numerical computation of the polynomial GCD as well as identification of nonisolated zeros of polynomial systems.

Key words. matrix, rank, rank-revealing, null space, singular value, updating, downdating, GCD, nonisolated solution, polynomial system

AMS subject classifications. 12D05, 15A03, 15A18, 65F30, 65H10

DOI. 10.1137/S0895479803435282

1. Introduction.

The numerical rank determination arises in many applications that involve matrix computations, such as those discussed in a series of proceedings, *SVD and Signal Processing,* I, II, III [5, 13, 18]. While the singular value decomposition (SVD) is undoubtedly the most reliable method of determining the rank numerically, there are certain drawbacks. Among them, it is quite expensive when matrices become large. Moreover, it may not be able to take the matrix structure into account, and it is not easy to update or downdate when rows/columns are inserted or deleted. Alternative methods have been proposed, such as rank-revealing QR decomposition (RRQR) [2, 3, 4] and rank-revealing two-sided orthogonal decompositions (UTV, or URV/ULV) [6, 16, 17].

In this paper, a new rank-revealing algorithm is presented. For a given $m \times n$ matrix A, instead of calculating a decomposition that reveals the approximate rank, our method calculates the approximate rank and null space of A directly. We briefly outline the method as follows. Without loss of generality, we assume $m \geq n$, and let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ be singular values of A. Since the smallest singular value $\sigma_{\min} \equiv \sigma_n$ satisfies

$$\sigma_{\min} = \min_{\|\mathbf{x}\|_2=1} \| A\mathbf{x} \|_2,$$

the problem of finding $\sigma_{\min}$ can be converted to solving the overdetermined system

$$\begin{pmatrix} \tau \mathbf{x}^\top \\ A \end{pmatrix} \mathbf{x} = \begin{pmatrix} \tau \\ 0 \end{pmatrix}, \quad \text{where } \tau > \sigma_n, \tag{1.1}$$

*Received by the editors September 24, 2003; accepted for publication (in revised form) by H. A. van der Vorst August 5, 2004; published electronically May 6, 2005.
http://www.siam.org/journals/simax/26-4/43528.html

†Department of Mathematics, Michigan State University, East Lansing, MI 48824 (li@math.msu.edu). This author's research was supported in part by NSF grant DMS-0104009.

‡Department of Mathematics, Northeastern Illinois University, Chicago, IL 60625 (zzeng@neiu.edu). This author's research was supported in part by NSF grant DMS-0412003.

for its least squares solution $\mathbf{x}$. For this purpose, one may use the Gauss–Newton iteration [3]

$$\begin{cases} \mathbf{x}_{j+1} = \mathbf{x}_j - \begin{pmatrix} 2\tau\mathbf{x}_j^\top \\ A \end{pmatrix}^+ \begin{pmatrix} \tau\mathbf{x}_j^\top\mathbf{x}_j - \tau \\ A\mathbf{x}_j \end{pmatrix}, \\ \varsigma_{j+1} = \dfrac{\| A\mathbf{x}_{j+1} \|_2}{\| \mathbf{x}_{j+1} \|_2}, \quad j = 0, 1, \ldots. \end{cases} \tag{1.2}$$

Here and throughout, for an arbitrary matrix B of full column rank, B^+ stands for its pseudo-inverse. It can be shown that (Lemma 4.1 in section 4) the Gauss–Newton iteration in (1.2) is essentially the inverse iteration on the matrix $A^\top A$ without undesirable matrix multiplication. The global convergence of the iteration is therefore warranted, and $(\varsigma_j, \mathbf{x}_j)$ will converge to the singular pair $(\sigma_n, \mathbf{v}_n)$. In this article, unless otherwise mentioned, we always use "singular vector" to represent the *right* singular vector. After $\sigma_n = \sigma_{\min}$ is calculated along with its associated singular vector $\mathbf{v}_n$, the matrix

$$A_\varrho = \begin{pmatrix} \varrho\mathbf{v}_n^\top \\ A \end{pmatrix}, \quad \varrho \in \mathbb{R}, \tag{1.3}$$

has the same set of singular values along with the associated singular vectors as those of A except the smallest singular value σ_n of A is replaced by the singular value $\sqrt{\varrho^2 + \sigma_n^2}$ of A_ϱ with associated singular vector $\mathbf{v}_n$ (Corollary 5.2 in section 5). Therefore, if we choose $\varrho = \|A\|_F$, then the replacement $\sqrt{\varrho^2 + \sigma_n^2}$ becomes the largest singular value of A_ϱ. In the meantime, the second smallest singular value σ_{n-1} of A becomes the smallest one of A_ϱ, and iteration (1.2) for finding the smallest singular pair of A can be applied to A_ϱ to calculate the singular pair $(\sigma_{n-1}, \mathbf{v}_{n-1})$ of A. This process can be continued recursively to calculate as many singular values of A as desired in ascending order $\sigma_n \le \sigma_{n-1} \le \cdots$, along with their associated singular vectors $\mathbf{v}_n, \mathbf{v}_{n-1}, \ldots$. Once σ_k is larger than the prescribed threshold $\theta > 0$, we will admit k as the approximate rank of A and the computed $\mathbf{v}_{k+1}, \ldots, \mathbf{v}_n$ as an orthonormal basis for the approximate null space of A.

Our method has been implemented as a MATLAB package RankRev and applied to many applications. In section 7 we present comprehensive numerical results of our code compared with UTV Tools [6] and the MATLAB SVD function. To calculate the approximate rank and null space of a given matrix that has a low rank deficit, our code can be 20 times faster than the full SVD when the matrix size becomes very large. Compared with UTV Tools, our method seems to be more robust and accurate in general, especially when the singular value gap at the rank threshold is relatively small. Moreover, row/column updating and downdating in our method, elaborated in section 8, are quite simple and straightforward. Separate numerical results are presented in section 8.5 comparing our method with UTV Tools in this respect. While UTV Tools may return incorrect ranks in certain difficult cases, our code always produces accurate results on all the matrices tested.

While rank-revealing has a large variety of applications, the development of our algorithm follows the needs of two important applications which emerged recently: a stable numerical algorithm for the computation of the GCD of univariate polynomials and the identification of nonisolated numerical solutions of polynomial systems. The details of those applications will be illustrated in section 9.

2. Notation, terminology, and definitions. The terms rank, nullity, and null space are used in the *exact* sense as in common linear algebra textbooks. In *numerical* linear algebra, the *approximate rank*, also known as the *numerical rank*, has a specific meaning given in Definition 2.1 below. Since the approximate rank, approximate null space, and approximate nullity are important concepts in our discussion, to be more clear and concise we shall use the specific terms *approxi-rank*, *approxi-null space*, and *approxi-nullity* for those notions. The usual notation $\text{rank}(A)$ remains the exact rank of a matrix A.

Throughout this paper, matrices are denoted by upper-case letters such as A, B, Q, R, etc. Lower-case boldface letters like $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{x}$ represent column vectors. The notation $(\cdot)^\top$ denotes the transpose of a matrix or vector $(\cdot)$, and vector spaces are denoted by a boldface upper-case letters like $\mathbf{W}$ with $\mathbf{W}^\perp$ denoting its orthogonal complement.

The definition of approxi-rank was first given by Golub, Klema, and Stewart [7]. We shall use a somewhat simplified definition.

DEFINITION 2.1. *For a given threshold $\theta > 0$, a matrix $A \in \mathbb{R}^{m\times n}$ has* approxi-rank *k within θ, denoted by $\text{rank}_\theta(A) = k$, if k is the smallest rank of all matrices within a 2-norm distance θ of A. Namely,*

$$k = \min_{\|A-B\|_2 \le \theta} \{\text{rank}(B)\}. \tag{2.1}$$

In this case, we also say the approxi-nullity *of A within θ is $n - k$.*

Notice that the exact rank of a matrix may be considered the approxi-rank of the matrix within zero.

The minimum in (2.1) is attainable [7, 12]: For $\theta > 0$, let $A = U\Sigma V^\top$ be the SVD of A with singular values satisfying

$$\sigma_1 \ge \cdots \sigma_k > \theta \ge \sigma_{k+1} \ge \cdots \ge \sigma_n. \tag{2.2}$$

Let $A_k = U\Sigma_k V^\top$ with $\Sigma_k = \text{diag}\{\sigma_1, \ldots, \sigma_k, 0, \ldots, 0\}$; then $\|A - A_k\|_2 = \sigma_{k+1} \le \theta$ and $\text{rank}(A_k) = \text{rank}_\theta(A) = k$ (see [7]). Moreover, A_k is nearest to A (with respect to the 2-norm) with rank k. In other words, for

$$\breve{\sigma} = \inf\big\{\mu \mid \text{rank}_\mu(A) = k\big\}, \tag{2.3}$$

we have $\|A - A_k\|_2 = \breve{\sigma}$. Let

$$\hat{\sigma} = \sup\big\{\eta \mid \text{rank}_\eta(A) = k\big\}.$$

We call the ratio $\gamma = \hat{\sigma}/\breve{\sigma}$ the *approxi-rank gap*. The size of this gap strongly influences the difficulties in achieving the accuracy of rank-revealing computation as shown in numerical examples in sections 7 and 8.5. If the singular values of A and the threshold θ satisfy (2.2), then clearly $\hat{\sigma} = \sigma_k$ and $\breve{\sigma} = \sigma_{k+1}$. When $\text{rank}_\theta(A) = k$, we called the null space of A_k the *approxi-null space* of A within θ since A_k is the nearest matrix to A with rank k. Let $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be the singular vectors of A (and A_k) associated with singular values σ_j, $j = 1, \ldots, n$; the approxi-null space of A is spanned by $\{\mathbf{v}_{k+1}, \ldots, \mathbf{v}_n\}$. The approxi-nullity of A equals the dimension of the approxi-null space. Any vector $\mathbf{v}$ satisfying $\|A\mathbf{v}\|_2 \le \theta$ is called an *approxi-null vector* of A within θ.

3. The basic algorithm. As before, let $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$. We first establish the equivalence between finding the smallest singular value $\sigma_{\min} \equiv \sigma_n$ of A and solving the least squares problem of the quadratic system

$$\begin{pmatrix} \tau\mathbf{x}^\top \\ A \end{pmatrix} \mathbf{x} = \begin{pmatrix} \tau \\ 0 \end{pmatrix} \quad \text{with } \tau > \sigma_n. \tag{3.1}$$

PROPOSITION 3.1. *Let $\mathbf{u} \in \mathbb{R}^n$ be a vector satisfying*

$$\left\| \begin{pmatrix} \tau\mathbf{u}^\top \\ A \end{pmatrix} \mathbf{u} - \begin{pmatrix} \tau \\ 0 \end{pmatrix} \right\|_2^2 = \min_{x\in\mathbb{R}^n} \left\| \begin{pmatrix} \tau\mathbf{x}^\top \\ A \end{pmatrix} \mathbf{x} - \begin{pmatrix} \tau \\ 0 \end{pmatrix} \right\|_2^2$$

with a scaling factor $\tau > \sigma_n$. Then $\mathbf{u}$ is in the subspace $\mathbf{W}$ spanned by the singular vector(s) of A associated with the smallest singular value(s).

Proof. Let $A = U\Sigma V^\top$ be the SVD of A with orthogonal U and V. Let $\mathbf{z} = V^\top\mathbf{x}$ or $\mathbf{x} = V\mathbf{z}$, where $\mathbf{x} = (x_1, \ldots, x_n)^\top$ and $\mathbf{z} = (z_1, \ldots, z_n)^\top$. Let

$$f(x_1, \ldots, x_n) = \left\| \begin{pmatrix} \tau\mathbf{x}^\top \\ A \end{pmatrix} \mathbf{x} - \begin{pmatrix} \tau \\ 0 \end{pmatrix} \right\|_2^2 = \left\| \begin{pmatrix} \tau\mathbf{x}^\top\mathbf{x} - \tau \\ A\mathbf{x} \end{pmatrix} \right\|_2^2;$$

then

$$\begin{aligned} f(x_1, \ldots, x_n) &= \tau^2 \left(\mathbf{x}^\top\mathbf{x} - 1\right)^2 + \| A\mathbf{x} \|_2^2 = \tau^2 \left(\mathbf{z}^\top\mathbf{z} - 1\right)^2 + \| \Sigma\mathbf{z} \|_2^2 \\ &= \tau^2 \left(z_1^2 + \cdots + z_n^2 - 1\right)^2 + \sigma_1^2 z_1^2 + \cdots + \sigma_n^2 z_n^2 \equiv g(z_1, \ldots, z_n). \end{aligned}$$

Assume $g(\mathbf{z})$ reaches its minimum at $\mathbf{z} = \mathbf{y} \equiv (y_1, \ldots, y_n)^\top$. Then

$$\left.\frac{\partial g}{\partial z_j}\right|_{\mathbf{z}=\mathbf{y}} = 0, \quad j = 1, \ldots, n, \quad \text{i.e., } 4\tau^2 \left(y_1^2 + \cdots + y_n^2 - 1\right) y_j + 2\sigma_j^2 y_j = 0.$$

If $\mathbf{y} \neq 0$, let $J = \{1 \leq j \leq n \mid y_j \neq 0\}$. Then for $j \in J$, $\sigma_j^2 = 2\tau^2(1 - \sum_{l\in J} y_l^2)$. Hence, $\sigma_j^2 \leq 2\tau^2$, and $\sigma_j = \sigma$ for all $j \in J$ for certain $\sigma \in \{\sigma_1, \ldots, \sigma_n\}$ with $\sigma < \sqrt{2}\tau$. It follows that

$$\begin{aligned} g(y_1, \ldots, y_n) &= \tau^2 \left(\sum_{j\in J} y_j^2 - 1\right)^2 + \sum_{j\in J} \sigma_j^2 y_j^2 = \tau^2 \left(\sum_{j\in J} y_j^2 - 1\right)^2 + \sigma^2 \sum_{j\in J} y_j^2 \\ &= \tau^2 \left(\sum_{j\in J} y_j^2 - 1\right)^2 + \sigma^2 \left(\sum_{j\in J} y_j^2 - 1\right) + \sigma^2 \\ &= \tau^2 \left(-\frac{\sigma^2}{2\tau^2}\right)^2 + \sigma^2 \left(-\frac{\sigma^2}{2\tau^2}\right) + \sigma^2 = \sigma^2 - \frac{\sigma^4}{4\tau^2}. \end{aligned}$$

Therefore, the possible minimum values of $g(\mathbf{z})$ are $\sigma_j^2 - \frac{\sigma_j^4}{4\tau^2}$, $j = 1, \ldots, n$, and, perhaps, $g(0, \ldots, 0) = \tau^2$. Those values are all attainable since, for every singular pair $(\sigma_j, \mathbf{v}_j)$, letting $\mathbf{z} = sV^\top\mathbf{v}_j$ with $s^2 = 1 - \frac{\sigma_j^2}{2\tau^2}$ yields

$$g(\mathbf{z}) = \tau^2(s^2 - 1)^2 + \sigma_j^2 s^2 = \tau^2 \frac{\sigma_j^4}{4\tau^4} + \sigma_j^2 \left(1 - \frac{\sigma_j^2}{2\tau^2}\right) = \sigma_j^2 - \frac{\sigma_j^4}{4\tau^2}.$$

The function $h(\beta) = \beta^2 - \frac{\beta^4}{4\tau^2}$ is increasing in $[0, \tau]$, so

$$\min_{j=1,\ldots,n} \left\{ \sigma_j^2 - \frac{\sigma_j^4}{4\tau^2} \right\} = \sigma_n^2 - \frac{\sigma_n^4}{4\tau^2} \leq \sigma_n^2 < \tau^2$$

and $g(z_1, \ldots, z_n)$ reaches the minimum if $\sigma = \sigma_n$. Consequently, $\sigma_j = \sigma_n$ for all $j \in J$, and $\mathbf{u} = \sum_{j \in J} y_j \mathbf{v}_j$, where $\mathbf{v}_j$ is the singular vector associated with σ_j, $j = 1, \ldots, n$. □

Based on Proposition 3.1, the smallest singular value of A can be calculated via solving system (3.1) by the Gauss–Newton iteration [3]:

$$\begin{cases} \mathbf{x}_{j+1} = \mathbf{x}_j - \begin{pmatrix} 2\tau \mathbf{x}_j^\top \\ A \end{pmatrix}^+ \begin{pmatrix} \tau \mathbf{x}_j^\top \mathbf{x}_j - \tau \\ A\mathbf{x}_j \end{pmatrix}, \\ \varsigma_{j+1} = \dfrac{\| A\mathbf{x}_{j+1} \|_2}{\| \mathbf{x}_{j+1} \|_2}, \quad j = 0, 1, \ldots. \end{cases} \tag{3.2}$$

We shall prove in section 4 that the scalar sequence ς_j, $j = 1, 2, \ldots$, always converges to the smallest singular value $\sigma_{\min}$ of A. And if $\sigma_{\min}$ is a simple singular value, namely, $\sigma_{n-1} \neq \sigma_n$, then the vector sequences $\frac{1}{\varsigma_j} A\mathbf{x}_j$ and $\mathbf{x}_j$, $j = 1, 2, \ldots$, converge to the corresponding left and right singular vectors, respectively. When $\sigma_{\min}$ is not simple, ς_j still converges to $\sigma_{\min}$, while $\frac{1}{\varsigma_j} A\mathbf{x}_j$ and $\mathbf{x}_j$ converge into left and right singular subspaces associated with $\sigma_{\min}$.

When A has more than one zero singular values, the matrix $\left(\begin{smallmatrix} 2\tau \mathbf{x}_j^\top \\ A \end{smallmatrix}\right)$ becomes rank deficient and its pseudoinverse is undefined. While exact rank deficiency rarely happens in real computation, when it occurs, replacing A by $A+E$ with tiny $\|E\|_2$ will ensure the existence of the pseudoinverse. Such substitution has virtually no effect on the computing results. For details, see [8].

In the remainder of this paper, we shall frequently refer to the iteration (3.2) above as "applying the Gauss–Newton iteration on matrix A" for solving the least squares quadratic system in (3.1).

4. The convergence theory. The theory of the Gauss–Newton iteration warrants its local convergence under some restrictions, and the convergence rate is at least linear. The following lemma shows that the Gauss–Newton iteration (3.2) on the overdetermined quadratic system (3.1) is essentially the inverse iteration on the matrix $A^\top A$, and the convergence is therefore global.

LEMMA 4.1. *Let $A \in \mathbb{R}^{m \times n}$ be of full column rank, and let $\{\mathbf{x}_j\}$ be a vector sequence generated by iteration* (3.2). *Then there are constants c_j, $j = 0, 1, \ldots$, such that*

$$\mathbf{x}_{j+1} = c_j \left(A^\top A\right)^{-1} \mathbf{x}_j. \tag{4.1}$$

Proof. For simplicity, let $\mathbf{x}$ and $\mathbf{y}$ denote $\mathbf{x}_j$ and $\mathbf{x}_{j+1}$, respectively. Now,

$$\begin{aligned} \mathbf{y} &= \mathbf{x} - \begin{pmatrix} 2\tau \mathbf{x}^\top \\ A \end{pmatrix}^+ \begin{pmatrix} \tau \mathbf{x}^\top \mathbf{x} - \tau \\ A\mathbf{x} \end{pmatrix} \\ &= \mathbf{x} - \left[\left(2\tau \mathbf{x}, A^\top\right) \begin{pmatrix} 2\tau \mathbf{x}^\top \\ A \end{pmatrix} \right]^{-1} \left(2\tau \mathbf{x}, A^\top\right) \begin{pmatrix} \tau \mathbf{x}^\top \mathbf{x} - \tau \\ A\mathbf{x} \end{pmatrix} \\ &= \mathbf{x} - \left(4\tau^2 \mathbf{x}\mathbf{x}^\top + A^\top A\right)^{-1} \left[\left(2\tau^2 \mathbf{x}\mathbf{x}^\top + A^\top A\right) \mathbf{x} - 2\tau^2 \mathbf{x} \right] \end{aligned}$$

$$= \mathbf{x} - \left(4\tau^2\mathbf{x}\mathbf{x}^\top + A^\top A\right)^{-1}\left[\left(4\tau^2\mathbf{x}\mathbf{x}^\top + A^\top A\right)\mathbf{x} - 2\tau^2\mathbf{x}(\mathbf{x}^\top\mathbf{x}) - 2\tau^2\mathbf{x}\right]$$
$$= \left(4\tau^2\mathbf{x}\mathbf{x}^\top + A^\top A\right)^{-1} 2\tau^2\left(1 + \mathbf{x}^\top\mathbf{x}\right)\mathbf{x}.$$

This yields

$$\begin{aligned}\left(4\tau^2\mathbf{x}\mathbf{x}^\top + A^\top A\right)\mathbf{y} &= 2\tau^2\left(1 + \mathbf{x}^\top\mathbf{x}\right)\mathbf{x},\\ \left(A^\top A\right)\mathbf{y} &= \tau^2\left(2 + 2\mathbf{x}^\top\mathbf{x} - 4\mathbf{x}^\top\mathbf{y}\right)\mathbf{x},\\ \mathbf{y} &= 2\tau^2\left(1 + \mathbf{x}^\top\mathbf{x} - 2\mathbf{x}^\top\mathbf{y}\right)\left(A^\top A\right)^{-1}\mathbf{x}.\end{aligned} \tag{4.2}$$

So, $\mathbf{y} = c(A^\top A)^{-1}\mathbf{x}$ with $c = \frac{2\tau^2\,(1+\mathbf{x}^\top\mathbf{x})}{1+4\tau^2\,\mathbf{x}^\top(A^\top A)^{-1}\mathbf{x}}$. ∎

For a given matrix $A \in \mathbb{R}^{m\times n}$ and a threshold $\theta > 0$, we assume $\mathrm{rank}_\theta(A) = k$ and the singular values of A satisfy

$$\sigma_1 \geq \cdots \geq \sigma_k = \widehat{\sigma} > \theta \geq \breve{\sigma} = \sigma_{k+1} \geq \cdots \geq \sigma_n.$$

Then $\mathbf{W} = \mathrm{span}\{\mathbf{v}_{k+1}, \ldots, \mathbf{v}_n\}$ is the approxi-null space of A, where $\mathbf{v}_j$ is the singular vector associated with σ_j for $j = k+1, \ldots, n$. The orthogonal complement $\mathbf{W}^\perp$ of $\mathbf{W}$ is $\mathrm{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$, and every vector $\mathbf{z} \in \mathbb{R}^n$ can be written as $\mathbf{z} = \widehat{\mathbf{z}} + \breve{\mathbf{z}}$ with $\widehat{\mathbf{z}} \in \mathbf{W}^\perp$ and $\breve{\mathbf{z}} \in \mathbf{W}$. We say a sequence of nonzero vectors $\{\mathbf{z}_j\}$ converges into $\mathbf{W}$ if

$$\lim_{j\to\infty} \frac{\| \widehat{\mathbf{z}}_j \|_2}{\| \breve{\mathbf{z}}_j \|_2} = 0, \quad \| \breve{\mathbf{z}}_j \|_2 \neq 0, \quad j = 0, 1, \ldots.$$

The approxi-rank depends critically on the threshold $\theta > 0$ one chooses, and the approxi-rank gap $\gamma = \widehat{\sigma}/\breve{\sigma}$ dictates its computing difficulties. The following proposition ensures that the vector sequence $\{\mathbf{x}_j\}$ generated by iteration (3.2) converges into the approxi-null space of A.

PROPOSITION 4.2. *Suppose $A \in \mathbb{R}^{m\times n}$ and $\mathrm{rank}_\theta(A) = k$ with a nontrivial approxi-null space $\mathbf{W}$ and approxi-rank gap γ. Then for $\mathbf{x}_0$ not orthogonal to $\mathbf{W}$, the iteration* (3.2) *generates a vector sequence $\{\mathbf{x}_j\}$ and a scalar sequence $\{\varsigma_j\}$, where $\mathbf{x}_j$ converges into $\mathbf{W}$ linearly in the sense*

$$\frac{\left\| \widehat{\mathbf{x}}_j \right\|_2}{\left\| \breve{\mathbf{x}}_j \right\|_2} \leq \left(\frac{1}{\gamma}\right)^{2j} \frac{\left\| \widehat{\mathbf{x}}_0 \right\|_2}{\left\| \breve{\mathbf{x}}_0 \right\|_2}, \quad j = 0, 1, \ldots, \tag{4.3}$$

and ς_j satisfies

$$\sigma_n \leq \varsigma_j \leq \breve{\sigma} + \left(\frac{1}{\gamma}\right)^{2j} \frac{\left\| \widehat{\mathbf{x}}_0 \right\|_2}{\left\| \breve{\mathbf{x}}_0 \right\|_2}\sigma_1. \tag{4.4}$$

Proof. Let $\mathbf{x}_0 = u_1\mathbf{v}_1 + \cdots + u_n\mathbf{v}_n$. From Lemma 4.1,

$$\mathbf{x}_1 = \eta\left(\frac{u_1}{\sigma_1^2}\mathbf{v}_1 + \cdots + \frac{u_n}{\sigma_n^2}\mathbf{v}_n\right)$$

for certain $\eta \in \mathbb{R}$, and with $\alpha = \frac{\eta}{\breve{\sigma}^2}$,

$$\mathbf{x}_1 = \alpha\left(\frac{\breve{\sigma}^2}{\sigma_1^2}u_1\mathbf{v}_1 + \cdots + \frac{\breve{\sigma}^2}{\sigma_n^2}u_n\mathbf{v}_n\right) = \widehat{\mathbf{x}}_1 + \breve{\mathbf{x}}_1,$$

where

$$\left\| \widehat{\mathbf{x}}_1 \right\|_2 = \left\| \alpha \left(\frac{\breve{\sigma}^2}{\sigma_1^2} u_1 \mathbf{v}_1 + \cdots + \frac{\breve{\sigma}^2}{\sigma_k^2} u_k \mathbf{v}_k \right) \right\|_2 \le |\alpha| \left(\frac{1}{\gamma} \right)^2 \left\| \widehat{\mathbf{x}}_0 \right\|_2,$$
$$\left\| \breve{\mathbf{x}}_1 \right\|_2 = \left\| \alpha \left(\frac{\breve{\sigma}^2}{\sigma_{k+1}^2} u_{k+1} \mathbf{v}_{k+1} + \cdots + \frac{\breve{\sigma}^2}{\sigma_n^2} u_n \mathbf{v}_n \right) \right\|_2 \ge |\alpha| \left\| \breve{\mathbf{x}}_0 \right\|_2.$$

Since $\| \breve{\mathbf{x}}_0 \|_2 \neq 0$, we have $\frac{\| \widehat{\mathbf{x}}_1 \|_2}{\| \breve{\mathbf{x}}_1 \|_2} \le (\frac{1}{\gamma})^2 \frac{\| \widehat{\mathbf{x}}_0 \|_2}{\| \breve{\mathbf{x}}_0 \|_2}$. By a simple induction, inequality (4.3) follows. For inequality (4.4),

$$\begin{aligned} \sigma_n &\le \frac{\| A\mathbf{x}_j \|_2}{\| \mathbf{x}_j \|_2} \le \frac{\left\| A \widehat{\mathbf{x}}_j \right\|_2}{\| \mathbf{x}_j \|_2} + \frac{\left\| A \breve{\mathbf{x}}_j \right\|_2}{\| \mathbf{x}_j \|_2} \\ &\le \left\| A \left(\frac{\widehat{\mathbf{x}}_j}{\left\| \breve{\mathbf{x}}_j \right\|_2} \right) \right\|_2 + \left\| A \left(\frac{\breve{\mathbf{x}}_j}{\left\| \breve{\mathbf{x}}_j \right\|_2} \right) \right\|_2 \\ &\le \sigma_1 \left[\left(\frac{1}{\gamma} \right)^{2j} \frac{\left\| \widehat{\mathbf{x}}_0 \right\|_2}{\left\| \breve{\mathbf{x}}_0 \right\|_2} \right] + \breve{\sigma}. \qquad \square \end{aligned}$$

As an important special case, if there is a significant gap in magnitude between σ_{n-1} and σ_n, then the iteration (3.2) converges to σ_n and its associated singular vector $\mathbf{v}_n$.

COROLLARY 4.3. *If $\sigma_{n-1} > \sigma_n$ and $\mathbf{x}_0$ satisfies $\mathbf{x}_0^\top \mathbf{v}_n \neq 0$, then for each j the matrix $B_j = \begin{pmatrix} 2\tau \mathbf{x}_j^\top \\ A \end{pmatrix}$ in the Gauss–Newton iteration in* (3.2) *is of full rank with a well-defined pseudoinverse. Moreover, the sequences $\{\varsigma_j\}$ and $\{\frac{\mathbf{x}_j}{\| \mathbf{x}_j \|_2}\}$ converge to σ_n and $\mathbf{v}_n$, respectively, with*

$$\left\| \frac{\mathbf{x}_j}{\| \mathbf{x}_j \|_2} - \mathbf{v}_n \right\|_2 \le \left(\frac{\sigma_n}{\sigma_{n-1}} \right)^{2j} \left[1 + \left(\frac{\sigma_n}{\sigma_{n-1}} \right)^{2j} \right] \frac{\left\| \widehat{\mathbf{x}}_0 \right\|_2}{\left\| \breve{\mathbf{x}}_0 \right\|_2},$$
$$\left| \varsigma_j - \sigma_n \right| \le \left(\frac{\sigma_n}{\sigma_{n-1}} \right)^{2j} \sigma_1 \frac{\left\| \widehat{\mathbf{x}}_0 \right\|_2}{\left\| \breve{\mathbf{x}}_0 \right\|_2}, \quad j = 1, 2, \ldots.$$

Proof. Since $\sigma_{n-1} > \sigma_n \ge 0$, $A\mathbf{v}_j \neq 0$ for $j = 1, \ldots, n-1$. So, B_0 is of full rank because of the assumption $\mathbf{x}_0^\top \mathbf{v}_n \neq 0$. Similarly B_j is of full rank for all $j > 0$ since $\mathbf{x}_j^\top \mathbf{v}_n \neq 0$ from (4.3). The proof of the remaining assertions is a straightforward verification. $\square$

5. Computing the approxi-null space. The iteration (3.2) produces a vector $\mathbf{w}_1$ in the approxi-null space $\mathbf{W}$ of A. When the approxi-nullity of A is bigger than one, we may stack a scalar multiple of $\mathbf{w}_1^\top$ on top of A to form a new matrix B. We will show in this section that when iteration (3.2) is applied to B it may produce another approxi-null vector $\mathbf{w}_2$ of A that is orthogonal to $\mathbf{w}_1$. This deflation-iteration process can be continued recursively to produce an orthonormal basis for the approxi-null space $\mathbf{W}$.

PROPOSITION 5.1. *Under the same assumptions of Proposition* 4.2*, for any unit vector* $\mathbf{w} \in \mathbf{W}$*, the matrix*

$$B = \begin{pmatrix} \varrho \mathbf{w}^\top \\ A \end{pmatrix} \quad \text{with } \varrho \geq \widehat{\sigma} \tag{5.1}$$

has singular values $\{\sigma'_j\}_{j=1}^n$ *satisfying*

$$\sigma'_1 \geq \cdots \geq \sigma'_{k+1} \geq \widehat{\sigma} > \breve{\sigma} \geq \sigma'_{k+2} \geq \cdots \geq \sigma'_n, \tag{5.2}$$

and its approxi-null space $\mathbf{W}'$ *spanned by the singular vectors of* B *associated with* $\sigma'_{k+2}, \ldots, \sigma'_n$ *is a subspace of* $\mathbf{W}$*.*

Proof. Since $\mathbf{w} \in \mathbf{W}$, we can write $\mathbf{w} = \rho_{k+1}\mathbf{v}_{k+1} + \cdots + \rho_n \mathbf{v}_n$ with $\rho_{k+1}^2 + \cdots + \rho_n^2 = 1$. The SVD $A = U\Sigma V^\top$ yields

$$\begin{aligned}
&\begin{pmatrix} 1 & \\ & U^\top \end{pmatrix} BV \\
&\quad = \begin{pmatrix} 0 & \cdots & 0 & \varrho\rho_{k+1} & \cdots & \varrho\rho_n \\ \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_k & & & \\ & & & \sigma_{k+1} & & \\ & & & & \ddots & \\ & & & & & \sigma_n \\ & & & & & \end{pmatrix} = P \begin{pmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_k & & & \\ & & & \varrho\rho_{k+1} & \cdots & \varrho\rho_n \\ & & & \sigma_{k+1} & & \\ & & & & \ddots & \\ & & & & & \sigma_n \end{pmatrix} \\
&\quad = P \begin{pmatrix} I_{k\times k} & \\ & \hat{U} \end{pmatrix} \begin{pmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_k & & & \\ & & & \hat{\sigma}_{k+1} & & \\ & & & & \ddots & \\ & & & & & \hat{\sigma}_n \end{pmatrix} \begin{pmatrix} I_{k\times k} & \\ & \hat{V}^\top \end{pmatrix},
\end{aligned}$$

where P is a permutation matrix with $\hat{U}$ and $\hat{V}$ being orthogonal matrices in the SVD of

$$D = \begin{pmatrix} \varrho\rho_{k+1} & \cdots & \varrho\rho_n \\ \sigma_{k+1} & & \\ & \ddots & \\ & & \sigma_n \end{pmatrix} = \hat{U} \begin{pmatrix} \hat{\sigma}_{k+1} & & \\ & \ddots & \\ & & \hat{\sigma}_n \end{pmatrix} \hat{V}^\top.$$

We claim that

$$\hat{\sigma}_{k+1} \geq \varrho \quad \text{and} \quad \hat{\sigma}_j \leq \sigma_{j-1}, \quad j = k+2, \ldots, n. \tag{5.3}$$

In fact, $\hat{\sigma}_{k+1}$ is the largest singular value of D which is larger than or equal to ϱ since

$$\hat{\sigma}_{k+1} = \max_{\mathbf{x}\in\mathbb{R}^{n-k},\, \|\mathbf{x}\|_2=1} \| D\mathbf{x} \|_2 \geq \left\| D \begin{pmatrix} \rho_{k+1} \\ \vdots \\ \rho_n \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} \varrho \\ \rho_{k+1}\sigma_{k+1} \\ \vdots \\ \rho_n\sigma_n \end{pmatrix} \right\|_2 \geq \varrho.$$

On the other hand, let $\mathbf{y} = (0, \ldots, 0, y_{n-1}, y_n)^\top \in \mathbb{R}^{n-k}$ such that $\| \mathbf{y} \|_2 = 1$ and $y_{n-1}\rho_{n-1} + y_n\rho_n = 0$. Then

$$\hat{\sigma}_n = \min_{\mathbf{x}\in\mathbb{R}^{n-k},\, \|\mathbf{x}\|_2=1} \| D\mathbf{x} \|_2 \leq \| D\mathbf{y} \|_2 = \sqrt{(\sigma_{n-1}y_{n-1})^2 + (\sigma_n y_n)^2} \leq \sigma_{n-1}.$$

Denote the columns of $\hat{V}$ by $\hat{\mathbf{v}}_{k+1}, \ldots, \hat{\mathbf{v}}_n$. For any fixed $j \in \{k+1, \ldots, n-2\}$, let $\mathbf{z} = (0, \ldots, 0, z_j, \ldots, z_n)^\top$ with $\| \mathbf{z} \|_2 = 1$, where $\hat{\mathbf{v}}_l^\top \mathbf{z} = 0$ for $l = j+2, \ldots, n$, and $\Sigma_{i=j}^n \rho_i z_i = 0$. Then

$$\begin{aligned} \hat{\sigma}_{j+1} &= \min \left\{ \| D\mathbf{x} \|_2 \mid \|\mathbf{x}\|_2 = 1, \mathbf{x}^\top \hat{\mathbf{v}}_l = 0, l = j+2, \ldots, n \right\} \\ &\leq \| D\mathbf{z} \|_2 = \sqrt{(z_j\sigma_j)^2 + \cdots + (z_n\sigma_n)^2} \leq \sigma_j \end{aligned}$$

and inequalities (5.3) hold. Consequently, they lead to the validity of the inequalities in (5.2) with

$$\left\{\sigma'_1, \ldots, \sigma'_{k+1}\right\} = \left\{\sigma_1, \ldots, \sigma_k, \hat{\sigma}_{k+1}\right\}, \quad \sigma'_l = \hat{\sigma}_l, \quad l = k+2, \ldots, n. \qquad \square$$

In practice, we may choose $\varrho = \|A\|_\infty$. In applying iteration (3.2) on B, as the least squares solution of

$$\begin{pmatrix} \tau\mathbf{w}_2^\top \\ \varrho\mathbf{w}_1^\top \\ A \end{pmatrix} \mathbf{w}_2 = \begin{pmatrix} \tau \\ 0 \\ 0 \end{pmatrix},$$

$\mathbf{w}_2 \in \mathbf{W}$, is approximately orthogonal to $\mathbf{w}_1$. Continuing this process recursively, an orthonormal basis for $\mathbf{W}$ can be constructed.

As an important special case, if $\sigma_{n-1} \gg \sigma_n$, iteration (3.2) converges to $\mathbf{w} = \mathbf{v}_n$ and $\varsigma = \sigma_n$. In this case, stacking $\varrho\mathbf{v}_n^\top$ on top of A makes σ_{n-1} the smallest singular value of the resulting matrix.

COROLLARY 5.2. *Let σ be a singular value of A with associated singular vector $\mathbf{v}$. The matrix*

$$A_\rho = \begin{pmatrix} \rho\mathbf{v}^\top \\ A \end{pmatrix} \tag{5.4}$$

has the same singular values and corresponding singular vectors as those of A, except the singular value σ of A is replaced by the singular value $\sqrt{\rho^2 + \sigma^2}$ of A_ρ associated with the same singular vector $\mathbf{v}$.

Proof. For simplicity, let $\sigma = \sigma_n$ and $A = U\Sigma V^\top$ be the SVD of A. We have

$$\begin{pmatrix} 1 & \\ & U^\top \end{pmatrix} \begin{pmatrix} \rho \mathbf{v}^\top \\ A \end{pmatrix} V = \begin{pmatrix} \rho \mathbf{v}^\top V \\ U^\top A V \end{pmatrix} = \begin{pmatrix} 0 & \cdots & 0 & \rho \\ \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_{n-1} & \\ & & & \sigma \\ & & & \end{pmatrix}.$$

By applying a Givens transformation from the left on ρ and σ, it is clear that the singular value σ of A is replaced by the singular value $\sqrt{\rho^2 + \sigma^2}$ of A_ρ, while the associated singular vector remains the same. ∎

6. The overall algorithm. As mentioned before, the approxi-rank k of matrix A depends critically on the chosen threshold $\theta > 0$ for which singular values of A satisfy

$$\sigma_1 \geq \cdots \geq \sigma_k > \theta > \sigma_{k+1} \geq \cdots \geq \sigma_n. \tag{6.1}$$

There is no uniform threshold for all applications. The user must make a decision on the threshold $\theta > 0$ based on the nature of the application.

The approxi-rank gap $\gamma = \frac{\sigma_k}{\sigma_{k+1}}$ may be considered a condition number for this rank-revealing problem. If γ is large, say, 10^3, then every iterative step in (3.2) will improve the convergence by six digits because in Proposition 4.2 the sequences $\{\mathbf{x}_j\}$ and $\{\varsigma_j\}$ satisfy

$$\frac{\left\| \widehat{\mathbf{x}}_j \right\|_2}{\left\| \widecheck{\mathbf{x}}_j \right\|_2} \leq \left(10^{-3}\right)^{2j} \frac{\left\| \widehat{\mathbf{x}}_0 \right\|_2}{\left\| \widecheck{\mathbf{x}}_0 \right\|_2} \quad \text{and} \quad \sigma_n \leq \varsigma_j \leq \sigma_{k+1} + \left(10^{-3}\right)^{2j} \frac{\left\| \widehat{\mathbf{x}}_0 \right\|_2}{\left\| \widecheck{\mathbf{x}}_0 \right\|_2} \sigma_1.$$

After three iteration steps they become

$$\frac{\left\| \widehat{\mathbf{x}}_3 \right\|_2}{\left\| \widecheck{\mathbf{x}}_3 \right\|_2} \leq 10^{-18} \frac{\left\| \widehat{\mathbf{x}}_0 \right\|_2}{\left\| \widecheck{\mathbf{x}}_0 \right\|_2} \quad \text{and} \quad \sigma_n \leq \varsigma_3 \leq \sigma_{k+1} + 10^{-18} \frac{\left\| \widehat{\mathbf{x}}_0 \right\|_2}{\left\| \widecheck{\mathbf{x}}_0 \right\|_2} \sigma_1.$$

Since the machine epsilon of IEEE standard double precision is about 2.2×10^{-16}, in this case $\mathbf{x}_3$ is sufficiently accurate to be an approxi-null vector unless the randomly chosen initial vector $\mathbf{x}_0$ is almost orthogonal to the approxi-null space.

Let $(\sigma_1, \mathbf{v}_1), \ldots, (\sigma_n, \mathbf{v}_n)$ be the singular pairs of A with σ_j's satisfying (6.1). For an input threshold $\theta > 0$, our algorithm begins with calculating the smallest singular pair $(\hat{\sigma}_n, \hat{\mathbf{v}}_n)$. If $\hat{\sigma}_n > \theta$, A will be classified as being of full approxi-rank, and the process stops. Otherwise, the algorithm continues by calculating singular pairs $(\hat{\sigma}_{n-1}, \hat{\mathbf{v}}_{n-1})$, $(\hat{\sigma}_{n-2}, \hat{\mathbf{v}}_{n-2})$, ... Once we reach $\hat{\sigma}_k > \theta$, the process will be terminated with k being the approxi-rank of A and $\text{span}\{\hat{\mathbf{v}}_{k+1}, \ldots, \hat{\mathbf{v}}_n\}$ the approxi-null space. If the approxi-rank gap $\gamma = \frac{\hat{\sigma}_k}{\hat{\sigma}_{k+1}}$ is not as large, it may need more than three iteration steps in (3.2) for each singular value. The users can set the number of iteration steps based on the nature of the application. The overall algorithm RankRev is shown in a pseudocode in Figure 6.1.

Pseudocode RankRev:
input: Matrix $A \in \mathbb{R}^{m\times n}$, threshold $\theta > 0$
output: approxi-rank k, orthonormal basis $\{\mathbf{w}_{k+1}, \ldots, \mathbf{w}_n\}$
for the approxi-null space.

```
Compute the QR decomposition A = QR
Initialize B = R,   τ = ||A||_∞
For k = n, n−1, ..., 1 do
    generate a random unit vector x_0
    for  j = 0, 1, 2  do
        D = [ 2τ x_j^T ; B ],  b = [ τ x_j^T x_j − τ ; B x_j ]
        Hessenberg QR decomposition D = QR
        backward substitution to solve Rz = Q^T b for z
        x_{j+1} = x_j − z,   ς = ||Rx_{j+1}||_2 ||x_{j+1}||_2^{-1}
        if ς < θ then
            w_k = x_{j+1} / ||x_{j+1}||_2
            break j-loop
        end if
    end do
    if ς > θ then
        break k-loop
    else
        Hessenberg QR decomposition [τ w_k^T ;  B] = QR
        update B = R
    end if
end do
```

FIG. 6.1. *Pseudocode of RankRev.*

Practically, the iteration (3.2) is carried out by finding a least squares solution $\Delta\mathbf{x}$ $(= \mathbf{x}_{j+1} - \mathbf{x}_j)$ to the linear system

$$\begin{pmatrix} 2\tau\mathbf{x}_j^\top \\ A \end{pmatrix} \Delta\mathbf{x} = -\begin{pmatrix} \tau\left(\mathbf{x}_j^\top\mathbf{x}_j - 1\right) \\ A\mathbf{x}_j \end{pmatrix} \tag{6.2}$$

at the jth stage. To avoid unnecessary QR decomposition of the full matrix at each step, we may calculate the QR decomposition of A *before* the iteration and update the QR decomposition at each step.

With QR factorization $A = Q\binom{R}{0}$, finding the least squares solution to (6.2) is equivalent to solving the least squares problem of

$$\begin{pmatrix} 2\tau\mathbf{x}_j^\top \\ R \end{pmatrix} \Delta\mathbf{x} = -\begin{pmatrix} \tau\mathbf{x}_j^\top\mathbf{x}_j - \tau \\ R\mathbf{x}_j \end{pmatrix} \tag{6.3}$$

for $\Delta\mathbf{x} = \mathbf{x}_{j+1} - \mathbf{x}_j$, in which one uses the QR decomposition of the upper Hessenberg matrix $\binom{2\tau\mathbf{x}_j^\top}{R}$. Updating the QR factorization of an n-column upper Hessenberg matrix requires n Givens transformations which cost $O(n^2)$ flops in total. After QR

updating, solving (6.3) for its least squares solution requires a total of $O(n^2)$ flops in backward substitutions.

The final QR factorization of $\binom{2\tau\mathbf{x}_j^\top}{R}$ can be used as the QR decomposition of the matrix B in (5.1) with $\rho = 2\tau$ and $\mathbf{w} = \mathbf{x}_j$. The computations are all on the order of $O(n^2)$ except the first QR factorization of A which costs $O(mn^2)$. Actually, on many occasions the QR decomposition of A had already been calculated for other purposes.

7. Numerical experiments and comparisons. Our rank-revealing algorithm is implemented as a MATLAB module RankRev that is electronically available from the authors upon request. Here we compare its efficiency, robustness, and accuracy with the MATLAB built-in SVD function as well as hurv in UTV Tools implemented by Fierro, Hansen, and Hansen [6]. The package UTV Tools is perhaps the only published comprehensive rank-revealing package with updating/downdating capabilities. All tests are carried out in MATLAB 6.1 on a Dell personal computer with a Pentium 4 CPU of 1.8 Mhz, 768 Mb of memory, and machine precision $\varepsilon \approx 10^{-16}$.

The main objective of our code RankRev is to calculate the approxi-rank and the approxi-null space of a matrix A that has a low approxi-nullity (equivalently, A is close to being of full approxi-rank) within a user-specified threshold. If the given matrix A is of approxi-rank about $n/2$, the full SVD can be more efficient. For low approxi-rank (i.e., high approxi-nullity) cases, UTV Tools function lurv and SVDPACK based on Lanczos method [1] are efficient options.

When $A \in \mathbb{R}^{m\times n}$ has an approxi-rank k within threshold $\theta > 0$, then A is often considered to be under a perturbation of a "noise" matrix E with $\|E\|_2 \le \theta$ such that $A - E$ has exact rank k. The 2-norm of E is often referred to as noise level. Usually, we consider a perturbation magnitude near machine precision, say, 1.0e-12, a low noise level, perturbation near 1, say, 1.0e-3, a high noise level, and the median noise level is around 1.0e-8.

7.1. Type 1: Low approxi-nullity, median noise level, small gap. Matrices for this test are of size $2n\times n$ with approxi-nullity fixed at 10 within threshold 10^{-8}. The singular values range from ε to $\|A\|_2 = 20$ with approxi-rank gap $\frac{\sigma_{n-10}}{\sigma_{n-9}} = 10^3$. Each matrix A is constructed using those specified singular values to form a diagonal matrix Σ and by setting $A = U\Sigma V^\top$ with randomly generated orthogonal matrices U and V with proper sizes. We use this type of matrix to test the efficiency and accuracy of RankRev compared with SVD and hurv for increasing n.

All three algorithms output accurate approxi-ranks. Table 7.1 lists the times and errors in executing SVD, hurv, and RankRev. The time measures are in seconds, and the error measures the distances of the computed approxi-null spaces to the spaces spanned by the right singular vectors associated with the ten smallest singular values. The results show that our RankRev is at least as efficient as hurv with significantly higher accuracy. When matrix sizes are in the thousands, both hurv

Table 7.1
Results for Type 1 *matrices.*

	Matrix sizes							
	400×200		800×400		1600×800		3200×1600	
	time	error	time	error	time	error	time	error
SVD	0.67	1e-15	5.6	2e-15	43.6	1e-15	1166.9	2e-15
hurv	1.41	1e-06	3.4	2e-06	11.6	1e-05	79.2	7e-06
RankRev	1.23	2e-09	3.3	2e-09	11.3	2e-09	48.8	2e-09

TABLE 7.2

Results for Type 2 *matrices. The computed approxi-ranks in parentheses are inaccurate results from* HURV.

	Matrix column size n	100	200	300	400	500
	Approxi-rank	50	100	150	200	250
HURV	Computed approxi-rank	50	100	150	(234)	(294)
	Approxi-null space error	1e-10	1e-05	3e-05	—	—
RANKREV	Computed approxi-rank	50	100	150	200	250
	Approxi-null space error	3e-10	6e-10	3e-08	5e-08	6e-08

TABLE 7.3

The accuracy measures on Type 3 *matrices without refinement. Due to the fixed size of the test matrices, the execution time is close to a constant for each code. We therefore list only the average time in the parentheses next to the code name.*

	Approxi-rank gaps γ					
Code (time)	10^6	10^5	10^4	10^3	10^2	10^1
HURV (4.86)	7.4e-11	2.7e-09	3.4e-08	1.6e-06	1.1e-04	1.8e-02
RANKREV (4.58)	7.4e-11	2.2e-10	6.3e-10	2.0e-09	6.9e-08	2.6e-04

and RANKREV are more than ten times faster than standard SVD even with the interpretation overhead in MATLAB codes.

7.2. Type 2: Median approxi-nullity, median noise level, small gap. Matrices used for this test are of $2n \times n$ with approxi-rank $\frac{n}{2}$ within threshold 10^{-8}. They are constructed in the same way as Type 1 above except for different singular values. The singular values range from ε to $\|A\|_2 = 20$ with approxi-rank gap $\gamma = 10^3$. While computing approxi-ranks of matrices of this sort is not the main goal of either RANKREV or HURV; we simply use them to test the robustness of both codes since both algorithms must recursively deflate $\frac{n}{2}$ times here. As shown in Table 7.2, the approxi-null space accuracy for HURV seems to deteriorate when n increases and it fails to provide accurate approxi-ranks for $n = 400$ and $n = 500$ even when its refinement option is activated.[1] In contrast, our code RANKREV always outputs accurate approxi-ranks and tiny errors in computed approxi-null spaces.

7.3. Type 3. Decreasing gaps, fixed size, low approxi-nullity, median noise level. Matrices A_j, $j = 6, 5, \ldots, 2, 1$, used in this test are of size 1000×500 with an approxi-nullity fixed at 10 within the same threshold 10^{-8}. The singular values range from ε to $\|A_j\|_2 = 20$. However, the approxi-rank gaps are set at 10^j for $j = 6, 5, \ldots, 2, 1$, respectively.

Table 7.3 lists the accuracy measures on computed approxi-null spaces with decreasing approxi-rank gaps. While the accuracy in computing the approxi-null spaces of both RANKREV and HURV deteriorate when the approxi-rank gap diminishes, our code RANKREV achieves a higher accuracy level with slightly faster speed. When tighter accuracy on the approxi-null space is required in application, while UTV Tools has its own refinement strategy [6, 11], we may simply set tighter criteria for stopping the Gauss–Newton iteration. Table 7.4 shows both codes are about equally accurate with their refinements.

[1] In a recent correspondence, the authors of UTV Tools indicated that the source of the problem leading to those failures has been identified and will be dealt with in future releases.

TABLE 7.4
The accuracy measures on Type 3 matrices with refinement.

Code (time)	Approxi-rank gaps					
	10^6	10^5	10^4	10^3	10^2	10^1
HURV (9.74)	7.4e-11	2.2e-10	6.3e-10	2.0e-09	6.9e-09	1.6e-08
RANKREV (7.82)	7.4e-11	2.2e-10	6.3e-10	2.0e-09	6.9e-09	1.8e-08

TABLE 7.5
Results for Type 4 matrices.

	Matrix sizes							
	400×200		800×400		1600×800		3200×1600	
	time	error	time	error	time	error	time	error
HURV	1.55	8.3e-05	3.42	3.2e-03	15.8	1.4e-04	71.9	2.6e-03
RANKREV	1.27	8.5e-11	3.05	6.8e-11	12.9	4.0e-10	48.5	1.6e-10

TABLE 7.6
Results for Type 5 matrices.

	Matrix sizes							
	400×200		800×400		1600×800		3200×1600	
	time	error	time	error	time	error	time	error
HURV	1.40	1.2e-15	3.47	9.0e-16	23.7	1.1e-15	—	failed
RANKREV	1.16	2.0e-14	3.11	4.1e-14	18.9	9.5e-14	53.7	6.5e-13

7.4. Type 4. High noise level, low approxi-nullity, small gap. The series of matrices used in this test are of $2n \times n$ with singular values in the interval $[1, 2]$ except ten small singular values in the interval $[0, 10^{-2}]$. Those matrices are used to test the accuracy and robustness of the rank-revealing computation in the presence of high noise level.

The results exhibited in Table 7.5 show the significant advantage of RANKREV over HURV in accuracy without refinement. If both codes activate the refinement option, however, HURV achieves slightly higher accuracy (2.9e-15) over RANKREV (4.3e-14), while RANKREV is slightly faster in speed by about 15%.

7.5. Type 5. Near-zero noise level, low approxi-nullity, large gap. This series of test matrices has singular values in the interval $[1, 2]$, except for the smallest ten, which are in the magnitude of machine precision. Table 7.6 shows that HURV consistently achieves slightly higher accuracy when the approxi-ranks are correctly determined, while our code maintains the advantage in efficiency. Nonetheless, HURV did not report an accurate approxi-rank for the 3200×1600 matrix. (The error appears to be machine-dependent. The authors of HURV are currently investigating it.)

8. Updating and downdating. For $A \in \mathbb{R}^{m \times n}$, the algorithm RANKREV in Figure 6.1 produces an approxi-rank k, a matrix $W = [\mathbf{w}_{k+1}, \mathbf{w}_{k+2}, \ldots, \mathbf{w}_n]$ whose columns form an orthonormal basis of the approxi-null space $\mathbf{W}$ of A, and a QR decomposition

$$\begin{pmatrix} \tau W^\top \\ A \end{pmatrix} = Q \begin{pmatrix} R \\ 0 \end{pmatrix}. \tag{8.1}$$

When a row/column is inserted in A, the determination of a new set of k, W, Q, and R of the new matrix using the information already available is called *updating*. It is called *downdating* if a row/column is deleted from A instead.

One of the main motivations in seeking alternatives to SVD in determining approxi-ranks is its difficulties in updating and downdating. The UTV decomposition possesses good updating capabilities, but its downdating seems somewhat complicated. In contrast, both updating and downdating in our method are straightforward and also quite stable and efficient.

In elaborating our procedure for updating and downdating, we shall repeatedly use the following QR downdating strategy [8, section 12.5.3].

We wish to compute the QR decomposition of the submatrix $\hat{B}$ in

$$B = \begin{bmatrix} \mathbf{b}^\top \\ \hat{B} \end{bmatrix}_{\begin{smallmatrix}1\\ m-1\end{smallmatrix}} = Q \begin{pmatrix} R \\ 0 \end{pmatrix} \in \mathbb{R}^{m\times n},$$

where the QR decomposition of B is available as given above. Let $\mathbf{q}^\top$ be the first row of Q and $G_1, \ldots, G_{m-1}$ be Givens rotations such that

$$G_1 \cdots G_{m-1}\mathbf{q} = \mathbf{e}_1.$$

Notice that

$$H = G_1 \cdots G_{m-1} \begin{pmatrix} R \\ 0 \end{pmatrix} = \begin{bmatrix} \mathbf{v}^\top \\ \hat{R} \\ 0 \end{bmatrix}_{\begin{smallmatrix}1\\ n\\ m-n-1\end{smallmatrix}}$$

is upper Hessenberg and

$$QG_{m-1}^\top \cdots G_1^\top = \begin{bmatrix} 1 & \\ & \hat{Q} \end{bmatrix}_{\begin{smallmatrix}1\\ m-1\end{smallmatrix}},$$

where $\hat{Q}$ is orthogonal. Thus, for $G = G_1 \cdots G_{m-1}$,

$$\begin{pmatrix} \mathbf{b}^\top \\ \hat{B} \end{pmatrix} = \left[QG^\top\right] \left[G \begin{pmatrix} R \\ 0 \end{pmatrix} \right] = \begin{bmatrix} 1 & \\ & \hat{Q} \end{bmatrix} \left[\begin{pmatrix} \mathbf{b}^\top \\ \hat{R} \\ 0 \end{pmatrix} \right], \tag{8.2}$$

and therefore

$$\hat{B} = \hat{Q} \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix}.$$

This QR downdating process requires $O(n^2)$ flops.

8.1. Column updating. For $A = (\mathbf{a}_1, \ldots, \mathbf{a}_n) \in \mathbb{R}^{m\times n}$ and $\mathbf{a}_{n+1} \in \mathbb{R}^m$, let $\hat{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_n, \mathbf{a}_{n+1})$. Clearly the approxi-null space $\hat{\mathbf{W}}$ of $\hat{A}$ contains $\{\hat{\mathbf{w}}_{k+1} \ldots, \hat{\mathbf{w}}_n\}$, where

$$\hat{\mathbf{w}}_j = \begin{pmatrix} \mathbf{w}_j \\ 0 \end{pmatrix}, \quad j = k+1, \ldots, n.$$

Those $\hat{\mathbf{w}}_j$'s remain orthonormal. The approxi-rank of $\hat{A}$ is either k or $k+1$. Only when it stays at k, the orthonormal basis of $\hat{\mathbf{W}}$ contains an additional vector which is the only approxi-null vector of the matrix

$$\breve{A} = \begin{pmatrix} \tau\hat{W}^\top \\ \hat{A} \end{pmatrix} = \begin{pmatrix} \tau W^\top & \mathbf{0} \\ A & \mathbf{a}_{n+1} \end{pmatrix}, \tag{8.3}$$

where $\hat{W} = [\hat{\mathbf{w}}_{k+1}, \ldots, \hat{\mathbf{w}}_n]$. For the QR decomposition

$$\begin{pmatrix} \tau W^\top \\ A \end{pmatrix} = Q \begin{pmatrix} R \\ 0 \end{pmatrix}$$

in (8.1), let

$$Q^\top \breve{A} = Q^\top \begin{pmatrix} \tau W^\top & \mathbf{0} \\ A & \mathbf{a}_{n+1} \end{pmatrix} = \begin{pmatrix} R & \mathbf{d}_1 \\ 0 & \mathbf{d}_2 \end{pmatrix}$$

and H be the Householder transformation satisfying

$$H\mathbf{d}_2 = (\zeta, 0, \ldots, 0)^\top.$$

Then

$$\breve{A} = \begin{pmatrix} \tau W^\top & \mathbf{0} \\ A & \mathbf{a}_{n+1} \end{pmatrix} = \left[Q \begin{pmatrix} I_{n\times n} & 0 \\ 0 & H^\top \end{pmatrix} \right] \begin{bmatrix} \begin{pmatrix} R & \mathbf{d}_1 \\ 0 & \zeta \end{pmatrix} \\ \begin{matrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{matrix} \end{bmatrix} = \tilde{Q} \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix}, \tag{8.4}$$

and we may obtain the possible additional approxi-null vector by

$$\begin{aligned} &\text{solving } R\mathbf{x} = -\mathbf{d}_1 \quad \text{for } \mathbf{x} \in \mathbb{R}^n \\ &\text{and setting } \mathbf{y} = \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}, \qquad \hat{\mathbf{w}}_{n+1} = \frac{1}{\|\mathbf{y}\|_2}\,\mathbf{y}. \end{aligned} \tag{8.5}$$

Clearly,

$$\hat{W}^\top \hat{\mathbf{w}}_{n+1} = 0 \quad \text{and} \quad \left\| \hat{A}\hat{\mathbf{w}}_{n+1} \right\|_2 = \frac{|\zeta|}{\|\mathbf{y}\|_2}.$$

When $\frac{|\zeta|}{\|\mathbf{y}\|_2}$ is below the threshold θ, $\hat{\mathbf{w}}_{n+1}$ becomes the additional approxi-null vector and $\{\hat{\mathbf{w}}_{k+1}, \ldots, \hat{\mathbf{w}}_{n+1}\}$ constitutes an orthonormal basis for the approxi-null space $\hat{\mathbf{W}}$ of $\hat{A}$.

For further updating or downdating, if needed, we also update the QR decomposition in (8.1):

$$\begin{pmatrix} \tau\hat{\mathbf{w}}_{n+1}^\top \\ \tau\hat{W}^\top \\ \hat{A} \end{pmatrix} = \begin{pmatrix} 1 & \\ & \tilde{Q} \end{pmatrix} \begin{bmatrix} \tau\hat{\mathbf{w}}_{n+1}^\top \\ \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix} \end{bmatrix} = \hat{Q} \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix}. \tag{8.6}$$

Computing $\hat{Q}$ and $\hat{R}$ requires $O(n^2)$ additional flops since $\tilde{R}$ is already upper-triangular.

If the new column is inserted between the $(l-1)$th and the lth column of A where $l < n$, we may first append the new column as the last (i.e., the $(n+1)$th) column and complete the computation described above. Then by switching its lth and $(n+1)$th components for each approxi-null vector $\hat{\mathbf{w}}_j$, $j = k+1, \ldots, n+1$, we obtain an orthonormal basis for the approxi-null space of $\hat{A}$, the new matrix with a new lth column inserted.

For further updating and/or downdating, the QR decomposition in (8.6) needs to be revised. We illustrate the process for $n = 4$ and $l = 2$ as

$$\begin{pmatrix} \tau\hat{W}^\top \\ \hat{A} \end{pmatrix} = \hat{Q} \begin{pmatrix} + & \times & + & + & + \\ & \times & + & + & + \\ & \times & & + & + \\ & \times & & & + \\ & \times & & & \end{pmatrix} = \hat{Q}G_1^\top \begin{pmatrix} + & \times & + & + & + \\ & \times & + & + & + \\ & \times & & + & + \\ & * & & & * \\ & 0 & & & * \end{pmatrix}$$

$$= \hat{Q}G_1^\top G_2^\top \begin{pmatrix} + & \times & + & + & + \\ & \times & + & + & + \\ & * & & * & * \\ & 0 & & * & * \\ & 0 & & & * \end{pmatrix} = \hat{Q}G_1^\top G_2^\top G_3^\top \begin{pmatrix} + & \times & + & + & + \\ & * & * & * & * \\ & 0 & * & * & * \\ & 0 & & * & * \\ & 0 & & & * \end{pmatrix}$$

$$= \check{Q} \begin{pmatrix} \check{R} \\ 0 \end{pmatrix}, \tag{8.7}$$

where the G_j's are the Givens rotations. The new $\check{Q}$ and $\check{R}$ are then available for further use.

We summarize the column updating process as follows.

- Input: matrix A, approxi-rank k, scaling factor τ, rank threshold θ, orthonormal basis for the approxi-null space $\mathbf{W}$, the QR decomposition Q and R as in (8.1), a new column $\mathbf{a}_{n+1}$, and its location l to be inserted.
- Append $\hat{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_n, \mathbf{a}_{n+1})$, form $\check{A}$ as in (8.3).
- Update the QR decomposition $\tilde{Q}$ and $\tilde{R}$ of $\check{A}$ as in (8.4).
- Calculate $\hat{\mathbf{w}}_{n+1}$ as in (8.5), and obtain the residual $\frac{|\zeta|}{\|\mathbf{y}\|_2}$.
- If the residual $\frac{|\zeta|}{\|\mathbf{y}\|_2} > \theta$, then
  - Set $k = k+1$, $\hat{W} = \left[\hat{\mathbf{w}}_{k+1}, \ldots, \hat{\mathbf{w}}_n\right]$, $\hat{Q} = \tilde{Q}$, $\hat{R} = \tilde{R}$

  else
  - Calculate $\hat{Q}$ and $\hat{R}$ as in (8.6), set $\hat{W} = \left[\hat{\mathbf{w}}_{k+1}, \ldots, \hat{\mathbf{w}}_n, \hat{\mathbf{w}}_{n+1}\right]$

  end if
- If $l \neq n+1$, then
  - Swap the lth and the $(n+1)$th components of every approxi-null vector as columns of $\hat{W}$
  - Calculate $\check{Q}$ and $\check{R}$ as in (8.7), set as $\hat{Q}$ and $\hat{R}$, respectively.

end if

- Output updated k, $\hat{W}$, $\hat{Q}$, $\hat{R}$.

While the only significant cost of updating is solving an upper-triangular system $R\mathbf{x} = -\mathbf{d}_1$ in (8.5) with $n^2 + O(n)$ flops when $\hat{Q}$ and $\hat{R}$ are not needed, the cost stays at $O(mn + n^2)$ with output $\hat{Q}$ and $\hat{R}$.

8.2. Column downdating. Let $\tilde{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_{l-1}, \mathbf{a}_{l+1}, \ldots, \mathbf{a}_n)$ where the lth colum $\mathbf{a}_l$ of A is deleted and $\tilde{\mathbf{W}}$ be its approxi-null space. If the approxi-nullity of A, or the dimension $n-k$ of its approxi-null space $\mathbf{W}$, is zero, then the approxi-nullity of $\tilde{A}$ remains zero, requiring no further computations. We therefore assume $n-k \geq 1$ and write

$$W = [\mathbf{w}_{k+1}, \ldots, \mathbf{w}_n] = \begin{pmatrix} w_{1,k+1} & \cdots & w_{1,n} \\ \vdots & \ddots & \vdots \\ w_{n,k+1} & \cdots & w_{n,n} \end{pmatrix} \in \mathbb{R}^{n \times (n-k)}.$$

Let $H \in \mathbb{R}^{(n-k)\times(n-k)}$ be the Householder transformation satisfying

$$H \begin{pmatrix} w_{l,k+1} \\ w_{l,k+2} \\ \vdots \\ w_{l,n} \end{pmatrix} = \begin{pmatrix} \eta \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{8.8}$$

This yields

$$WH^\top = [\hat{\mathbf{w}}_{k+1}, \ldots, \hat{\mathbf{w}}_n] = \begin{pmatrix} * & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & * \\ \eta & 0 & \cdots & 0 \\ * & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & \cdots & * & * \end{pmatrix} \longleftarrow l\text{th row.}$$

Because

$$\left(WH^\top\right)^\top \left(WH^\top\right) = H\left(W^\top W\right)H^\top = H^\top I_{n-k} H = I_{n-k},$$

the columns of $WH^\top$ also form an orthonormal basis for $\mathbf{W}$. By removing the lth component of $\hat{\mathbf{w}}_j$ for each $j = k+1, \ldots, n$, we obtain a set of vectors $\tilde{\mathbf{w}}_{k+1}, \ldots, \tilde{\mathbf{w}}_n$ satisfying

$$\tilde{A}\tilde{\mathbf{w}}_{k+1} = A\hat{\mathbf{w}}_{k+1} - \eta\,\mathbf{a}_l, \qquad \tilde{A}\tilde{\mathbf{w}}_j = A\hat{\mathbf{w}}_j, \quad j = k+2, \ldots, n.$$

Apparently, $\{\tilde{\mathbf{w}}_{k+2}, \ldots, \tilde{\mathbf{w}}_n\}$ is a subset of an orthonormal basis for $\tilde{\mathbf{W}}$, and the magnitude of $\| \tilde{A}\tilde{\mathbf{w}}_{k+1} \|_2$ determines the possible existence of an additional approxi-null vector: when it is small enough, the normalization of $\tilde{\mathbf{w}}_{k+1}$ completes $\{\tilde{\mathbf{w}}_{k+1}, \ldots, \tilde{\mathbf{w}}_n\}$ as an orthonormal basis for $\tilde{W}$.

To downdate the QR decomposition in (8.1) for further updating/downdating, since

$$\begin{pmatrix} \tau H\, W^\top \\ A \end{pmatrix} = \left[\begin{pmatrix} H & \\ & I \end{pmatrix} Q \right] \begin{pmatrix} R \\ 0 \end{pmatrix},$$

we have

$$\begin{pmatrix} \tau \tilde{W}^\top \\ \tilde{A} \end{pmatrix} = \left[\begin{pmatrix} H & \\ & I \end{pmatrix} Q \right] \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix} = \left[\begin{pmatrix} H & \\ & I \end{pmatrix} Q G_l^\top \cdots G_{n-1}^\top \right] \begin{pmatrix} \check{R} \\ 0 \end{pmatrix}, \tag{8.9}$$

where $\hat{R}$ is obtained from R by deleting its lth column and $G_l, \cdots G_{n-1}$ are the Givens rotations that transform $\hat{R}$ into upper-triangular $\check{R}$. Applying the QR downdating technique in (8.2) yields

$$\begin{pmatrix} \tau \tilde{\mathbf{w}}_{k+2}^\top \\ \vdots \\ \tau \tilde{\mathbf{w}}_n^\top \\ \tilde{A} \end{pmatrix} = \tilde{Q} \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix}. \tag{8.10}$$

The column downdating process stops here if $\tilde{\mathbf{w}}_{k+1}$ is not an approxi-null vector. Otherwise, we will stack $\tau \tilde{\mathbf{w}}_{k+1}^\top$ as the top row and update the QR decomposition in (8.10):

$$\begin{aligned} \begin{pmatrix} \tau \tilde{\mathbf{w}}_{k+1}^\top \\ \tau \tilde{\mathbf{w}}_{k+2}^\top \\ \vdots \\ \tau \tilde{\mathbf{w}}_n^\top \\ \tilde{A} \end{pmatrix} &= \begin{pmatrix} 1 & \\ & \tilde{Q} \end{pmatrix} \begin{pmatrix} \tau \tilde{\mathbf{w}}_n \\ \tilde{R} \\ 0 \end{pmatrix} \\ &= \left[\begin{pmatrix} 1 & \\ & \tilde{Q} \end{pmatrix} \begin{pmatrix} G^\top & \\ & I \end{pmatrix} \right] \begin{pmatrix} \breve{R} \\ 0 \end{pmatrix} = \breve{Q} \begin{pmatrix} \breve{R} \\ 0 \end{pmatrix}, \end{aligned} \tag{8.11}$$

where G is a product of $n-1$ Givens rotations that transforms the upper Hessenberg matrix $\begin{pmatrix} \tau \tilde{\mathbf{w}}_{k+1} \\ \tilde{R} \end{pmatrix}$ into upper triangular form $\breve{R}$.

The column downdating algorithm can be summarized as follows:

- Input: matrix A, approxi-rank k, scaling factor τ, threshold θ, orthonormal basis $\{\mathbf{w}_{k+1}, \ldots, \mathbf{w}_n\}$ for the approxi-null space $\mathbf{W}$, the QR decomposition Q and R as in (8.1), a column index l indicating the lth column of A is to be deleted.
- Form $W = [\mathbf{w}_{k+1}, \ldots, \mathbf{w}_n]$ and the Householder transformation H in (8.8).
- Set $\hat{W} = W\, H^\top$ and η as in (8.8).
- Get $\tilde{W} = [\tilde{\mathbf{w}}_{k+1}, \ldots, \tilde{\mathbf{w}}_n]$ by deleting the lth row of $\hat{W}$ and normalizing the first column afterward.
- Form the QR decomposition (8.9).
- Apply the QR downdating process (8.2) on (8.9) to obtain $\tilde{Q}$ and $\tilde{R}$ in (8.10).
- If $\| \tilde{A} \tilde{\mathbf{w}}_{k+1} \|_2 > \theta$, then

– Output k, $\tilde{W} = [\tilde{\mathbf{w}}_{k+2}, \ldots, \tilde{\mathbf{w}}_n]$, $\tilde{Q}$, $\tilde{R}$, the approxi-rank stays at k.

else

– Update the QR decomposition as in (8.11).
– Output $k = k-1$, $\tilde{W} = [\tilde{\mathbf{w}}_{k+1}, \ldots, \tilde{\mathbf{w}}_n]$, $\breve{Q}$, $\breve{R}$, the approxi-rank reduces by one.

end if

It requires $O(n^2)$ flops to carry out the column downdating process.

8.3. Row updating. Inserting a row $\mathbf{b}^\top$ into A for a new matrix $\hat{A}$, the approxi-rank of $\hat{A}$ will remain the same unless the approxi-rank k of A is less than n. In such cases, it is clear that the approxi-null space $\hat{\mathbf{W}}$ of $\hat{A}$ is a subset of the approxi-null space $\mathbf{W}$ of A, and they are equal if $\mathbf{b}$ is approximately orthogonal to $\mathbf{W}$. Namely, $\mathbf{W} = \hat{\mathbf{W}}$ if $\| W^\top \mathbf{b} \|_2 \le \theta$, where $W = [\mathbf{w}_{k+1}, \ldots, \mathbf{w}_n] \in \mathbb{R}^{n\times(n-k)}$, whose columns form an orthogonal basis of $\mathbf{W}$. When $\| W^\top \mathbf{b} \|_2 > \theta$, the approxi-rank of the new matrix $\hat{A}$ becomes $k+1$. To find an orthonormal basis of $\hat{\mathbf{W}}$ in this case, we first let $\mathbf{y} = W^\top \mathbf{b} \in \mathbb{R}^{n-k}$ and $H \in \mathbb{R}^{(n-k)\times(n-k)}$ be the Householder transformation such that $H\mathbf{y} = (\|\mathbf{y}\|_2, 0, \ldots, 0)^\top$. Denoting $H = [\mathbf{y}_{k+1}, \ldots, \mathbf{y}_n]$, we have $\{\mathbf{y}\}^\perp = \text{span}\{\mathbf{y}_{k+2}, \ldots, \mathbf{y}_n\}$. For $E = [\mathbf{y}_{k+2}, \ldots, \mathbf{y}_n] \in \mathbb{R}^{(n-k)\times(n-k-1)}$, let $WE = [\hat{\mathbf{w}}_{k+2}, \ldots, \hat{\mathbf{w}}_n] \in \mathbb{R}^{n\times(n-k-1)}$. The columns $\{\hat{\mathbf{w}}_{k+2}, \ldots, \hat{\mathbf{w}}_n\}$ form an orthonormal basis for $\hat{\mathbf{W}}$ because

$$(WE)^\top (WE) = E^\top \left(W^\top W\right) E = I_{(n-k-1)\times(n-k-1)},$$

and for $j = k+2, \ldots, n$, $\| \hat{A}\hat{\mathbf{w}}_j \|_2 = \| A\hat{\mathbf{w}}_j \|_2$ since $\mathbf{b}^\top W E = \mathbf{y}^\top E = 0$.

To update the QR decomposition in (8.1), we apply the QR downdating strategy in (8.2) on

$$\begin{pmatrix} \tau H W^\top \\ A \end{pmatrix} = \left[\begin{pmatrix} H & \\ & I \end{pmatrix} Q \right] \begin{pmatrix} R \\ 0 \end{pmatrix} = \breve{Q} \begin{pmatrix} R \\ 0 \end{pmatrix} \tag{8.12}$$

to delete its first row, yielding

$$\begin{pmatrix} \tau E^\top W^\top \\ A \end{pmatrix} = \hat{Q} \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix}. \tag{8.13}$$

When inserting a new row $\mathbf{b}^\top$ into A, let P be the permutation matrix that swaps the new row to the top. It follows that

$$\begin{pmatrix} \tau E^\top W^\top \\ \hat{A} \end{pmatrix} = P \begin{pmatrix} \mathbf{b}^\top \\ \tau E^\top W^\top \\ A \end{pmatrix} = P \begin{pmatrix} 1 & \\ & \breve{Q} \end{pmatrix} \begin{pmatrix} \mathbf{b}^\top \\ \hat{R} \\ 0 \end{pmatrix} = \tilde{Q} \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix}, \tag{8.14}$$

where

$$\tilde{Q} = P \begin{pmatrix} 1 & \\ & \breve{Q} \end{pmatrix} G^\top, \qquad G \begin{pmatrix} \mathbf{b}^\top \\ \hat{R} \\ 0 \end{pmatrix} = \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix},$$

and G is the product of n Givens rotations.

Our row-updating algorithm can be summarized as follows:

- Input: matrix A, approxi-rank k, scaling factor τ, threshold θ, orthonormal basis $\{\mathbf{w}_{k+1}, \ldots, \mathbf{w}_n\}$ for the approxi-null space $\mathbf{W}$, the QR decomposition Q and R as in (8.1), a new row $\mathbf{b}^\top$, and the row index l indicating $\mathbf{b}^\top$ will be inserted above the lth row of A.
- Form $W = [\mathbf{w}_{k+1}, \ldots, \mathbf{w}_n]$.
- If $\| W^\top \mathbf{b} \|_2 < \theta$, then
  - Update the QR decomposition (8.1) for inserting $\mathbf{b}^\top$
  - Output k, W and the updated Q, R.

  else
  - Construct the Householder transformation H such that $H(W^\top \mathbf{b}) = (*, 0, \ldots, 0)^\top$.
  - Use H to get $\breve{Q}$ as in (8.12).
  - Downdate the QR decomposition (8.12) to obtain $\hat{Q}$ and $\hat{R}$ in (8.13)
  - Insert $\mathbf{b}^\top$ into A and update the QR decomposition (8.13) to obtain $\tilde{Q}$ and $\tilde{R}$ in (8.14)
  - Output $k = k + 1$, $\tilde{W} = WE$, $\tilde{Q}$, $\tilde{R}$, the approxi-rank increases by one.

  end if

8.4. Row downdating. Let $\breve{A}$ be the matrix obtained from A by deleting its lth row $\mathbf{r}^\top$. For a proper permutation matrix P, we have, from (8.1),

$$\begin{pmatrix} \mathbf{r}^\top \\ \tau W^\top \\ \breve{A} \end{pmatrix} = P \begin{pmatrix} \tau W^\top \\ A \end{pmatrix} = [PQ] \begin{pmatrix} R \\ 0 \end{pmatrix}. \tag{8.15}$$

Applying the QR downdating algorithm (8.2) on this QR decomposition yields

$$\tilde{A} = \begin{pmatrix} \tau W^\top \\ \breve{A} \end{pmatrix} = \breve{Q} \begin{pmatrix} \breve{R} \\ 0 \end{pmatrix}. \tag{8.16}$$

Obviously, the approxi-null space $\hat{\mathbf{W}}$ of $\hat{A}$ contains the approxi-null space $\mathbf{W}$ of A. For the possible emergence of an extra approxi-null vector of $\hat{A}$, we may apply the Gauss–Newton iteration (3.2) on the matrix $\breve{R}$ to calculate the singular vector. As explained earlier, if this singular vector is indeed an extra approxi-null vector, it is orthogonal to columns of W and forms an orthonormal basis for $\hat{\mathbf{W}}$ along with columns of W. We omit the pseudocode since the process is a straightforward application of QR downdating algorithm.

Remark. As mentioned in [6], row downdating may be difficult and complex for UTV decomposition: "... [W]e want to emphasize that numerically stable UTV downdating algorithms have become very complex, and the computational overhead can become quite large, especially when the exact rank decreases. It may be worth to consider whether recomputation of the ULV decomposition ... is to be preferred." In comparison, row downdating in our algorithm seems quite straightforward.

8.5. Numerical results on updating and downdating. Our updating and downdating algorithms have been thoroughly tested for all circumstances of inserting/deleting rows or columns. Since UTV Tools [6] contains only row updating and row downdating modules, we shall restrict our comparisons with UTV Tools to those situations only. The results of our method on column updating and downdating are quite similar.

TABLE 8.1
Comparisons on random row updating with changing approxi-ranks.

		Number of random rows inserted						
		1	2	3	···	8	9	10
Time (seconds)	URV_UP	0.66	0.64	0.63	···	0.67	0.67	0.49
	ROWUP	1.00	0.98	0.95	···	0.98	0.98	0.98
Approxi-null space accuracy	URV_UP	1e-8	1e-8	1e-8	···	3e-8	1e-8	0.0
	ROWUP	3e-9	2e-8	2e-8	···	4e-8	2e-9	0.0

The two modules in UTV Tools for row updating and row downdating are URV_UP and URV_DW, respectively. The updating module URV_UP works on inserting a row at the bottom, and the downdating module URV_DW applies to deleting the top row. Row inserting/deleting may or may not change the approxi-rank. Our tests show that there seems to be a significant difference in performance for both modules of UTV Tools in rank invariant and rank altering cases.

All tests in this section are conducted on the same computer listed in section 7. Both URV_UP and URV_DW are set to use their default control parameters, while our codes ROWUP and ROWDOWN are set to optimize the speed.

8.5.1. Row updating with changing approxi-ranks. The test matrix is initially a 1000×500 matrix having an approxi-nullity 10 with threshold 10^{-8}. The approxi-rank gap is $\gamma = 10^4$. After executing our RANKREV and HURV on this initial matrix separately, a random vector is inserted at the bottom in each updating step. Therefore, every update results in an increase in the approxi-rank by one. Both URV_UP in UTV Tools and our ROWUP have no difficulty identifying the increasing approxi-ranks with nearly identical accuracy in the updated approxi-null space. As shown in Table 8.1, URV_UP is considerably faster than our ROWUP in this case.

8.5.2. Row updating without changing approxi-ranks. When no changes in the approxi-rank occur for row updating, the code URV_UP in UTV Tools seems to have difficulties in identifying the approxi-ranks during the recursive updating, especially when the approxi-rank gap is not large enough. Even when the gap is large, URV_UP is still prone to miscalculating the approxi-rank at certain points. In contrast, our code ROWUP always outputs accurate approxi-ranks in all occasions and runs more than twice as fast.

Table 8.2 shows this event in a typical example. The initial matrix has the same features as the one in section 8.5.1 except the approxi-rank gap γ is increased to 10^6 since URV_UP fails too soon for the gap 10^4. A sequence of rows consisting of linear combinations of the existing rows are inserted at the bottom one at a time. The approxi-rank should stay at 490. However, after certain steps in the recursive updating, URV_UP outputs inaccurate approxi-ranks.

8.5.3. Row downdating without changing approxi-ranks. When deleting a row does not change the approxi-rank, our code ROWDOWN and its counterpart URV_DW in UTV Tools show similar performance in both efficiency and accuracy. The test starts by constructing an initial matrix $A \in \mathbb{R}^{1000\times 500}$ with the same features as in the initial matrix in section 8.5.1. Then 20 rows that are linear combinations of the existing rows of A are generated and stacked on top of A. Deleting those rows one by one does not alter the approxi-rank. Table 8.3 shows the results.

TABLE 8.2
Comparisons on random row updating without changing approxi-ranks. Data in parentheses indicate inaccurate computation.

		Number of linearly dependent rows inserted							
		1	2	···	5	6	7	···	10
Time (seconds)	URV_UP	1.09	1.14	···	1.11	0.69	0.69	···	1.11
	ROWUP	0.39	0.50	···	0.39	0.48	0.59	···	0.42
Approxi-null space error	URV_UP	2e-6	4e-6	···	3e-6	(0.15)	(0.06)	···	(0.14)
	ROWUP	1e-9	1e-9	···	2e-9	2e-9	3e-9	···	3e-9
Approxi-rank output	URV_UP	490	490	···	490	(491)	(492)	···	(492)
	ROWUP	490	490	···	490	490	490	···	490

TABLE 8.3
Comparisons on random row downdating without changing approxi-ranks.

		Number of linear dependent rows deleted						
		1	2	3	···	8	9	10
Time (seconds)	URV_DW	0.75	0.73	0.78	···	0.75	0.72	0.72
	ROWDOWN	0.78	0.78	0.89	···	0.76	0.70	0.70
Approxi-null space error	URV_DW	6e-8	1e-7	1e-7	···	4e-7	4e-7	4e-7
	ROWDOWN	6e-8	1e-7	1e-7	···	4e-7	4e-7	4e-7

8.5.4. Row downdating with decreasing approxi-ranks. As mentioned in [6], UTV decomposition may have difficulties in downdating especially when it reduces the approxi-ranks. This phenomenon does occur in the experiment we conducted below. We downdate a matrix of 1030×500 obtained by stacking 30 random rows on top of a matrix A of size 1000×500 with an approxi-nullity 30 within a threshold of 10^{-8}. The approxi-rank gap is set at a relatively large threshold 10^6. During the test, the 30 random rows are deleted one-by-one and both URV_DW and ROWDOWN are used to downdate the approxi-rank and the approxi-null space. The approxi-rank should decrease by one at every downdating step.

Table 8.4 shows that when downdating the approxi-rank accurately as in steps 1 to 15, both URV_DW and ROWDOWN exhibit similar efficiency and accuracy. At step 16, URV_DW miscalculates the approxi-rank by one and this error is carried on in remaining downdating steps, whereas our code ROWDOWN always produces the correct approxi-rank.

It is not clear whether the inaccurate outputs of UTV Tools in those difficult tests in both sections 7 and 8.5 are inherent in the UTV decomposition or the results of coding errors. They are under investigation by the authors of UTV Tools.

9. Applications.

9.1. Computing polynomial GCD. A new method for computing the GCD of univariate polynomials plays a key role in establishing a novel algorithm that accurately calculates polynomial roots and their multiplicities without using multiprecision arithmetic even if the polynomial is inexactly given [19]. This root-finding method is implemented in the MATLAB package MULTROOT [20]. Our rank-revealing method and recursive column updating constitute indispensable components in the new GCD finder and the root finder.

For any polynomial $h(x) = h_0x^k + h_1x^{k-1} + \cdots + h_k$, its coefficient vector is

TABLE 8.4
Comparisons on random row downdating with changing approxi-ranks. Data in parentheses indicate inaccurate computation.

		Number of linearly independent rows deleted						
		1	···	15	16	17	···	30
Approxi-null space error	URV_DW	9e-10	···	6e-10	(0.89)	(0.89)	···	(0.87)
	ROWDOWN	9e-10	···	2e-9	3e-9	3e-9	···	6e-8
Approxi-rank output	URV_DW	499	···	485	(485)	(484)	···	(471)
	ROWDOWN	499	···	485	484	483	···	470
Time (seconds)	URV_DW	1.06	···	1.17	—	—	···	—
	ROWDOWN	0.95	···	1.03	1.03	1.05	···	1.03

denoted by $\mathbf{h} = (h_0, h_1, \ldots, h_k)^\top$. Let $p(x)$ and $q(x)$ be polynomials of degrees n and m, respectively. Write

$$p(x) = p_0 x^n + p_1 x^{n-1} + \cdots + p_n,$$
$$q(x) = q_0 x^m + q_1 x^{m-1} + \cdots + q_m,$$

with $n \geq m$. A polynomial $u(x)$ is a GCD of $p(x)$ and $q(x)$, denoted by $\mathrm{GCD}\,(p, q)$, if there are polynomials $v(x)$ and $w(x)$ such that

$$\begin{aligned} u(x)v(x) &= p(x), \\ u(x)w(x) &= q(x), \end{aligned} \tag{9.1}$$

where $v(x)$ and $w(x)$ share no common roots (or, equivalently, no common factors). The $(n + r) \times (n - m + 2r)$ matrix

$$S_r(p,q) = \begin{pmatrix} \overbrace{\begin{matrix} p_0 & & \\ p_1 & \ddots & \\ \vdots & \ddots & p_0 \\ p_n & & p_1 \\ & \ddots & \vdots \\ & & p_n \end{matrix}}^{r} & \overbrace{\begin{matrix} q_0 & & \\ q_1 & \ddots & \\ \vdots & \ddots & q_0 \\ q_m & & q_1 \\ & \ddots & \vdots \\ & & q_m \end{matrix}}^{n-m+r} \end{pmatrix}, \quad n \geq m,\ r = 1, 2, \ldots, m,$$

is called the rth *Sylvester subresultant matrix* of p and q. The following results are well known [14, 19]:

- $S_m(p, q)$ is of rank $n + m - 2l + 1$ if and only if $\deg(\mathrm{GCD}\,(p, q)) = l \geq 1$.
- $S_r(p, q)$ has full rank if and only if $\deg(\mathrm{GCD}\,(p, q)) \leq m - r$.
- If $S_{r-1}(p, q)$ is of full rank and $S_r(p, q)$ is rank deficient, then the null space of $S_r(p, q)$ is spanned by the vector $\begin{pmatrix} \mathbf{w} \\ -\mathbf{v} \end{pmatrix}$ whose components $\mathbf{v} \in \mathbb{R}^{n-m+r}$ and $\mathbf{w} \in \mathbb{R}^r$ are coefficient vectors of $v(x)$ and $w(x)$ in (9.1), respectively. Also, $\deg(\mathrm{GCD}\,(p, q)) = m - r + 1$.

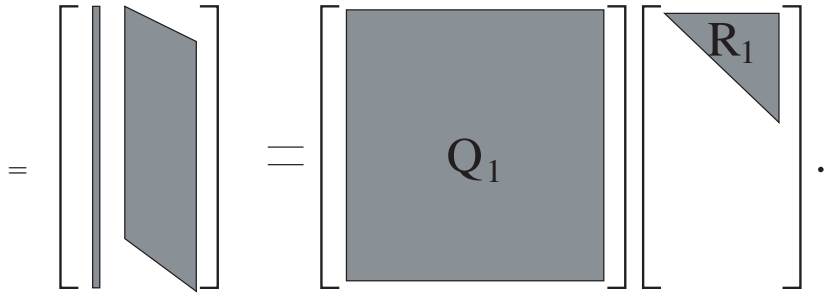
From those results, one may calculate $\mathrm{GCD}\,(p, q)$ by finding the first approxi-rank deficient Sylvester matrix $S_r(p, q)$ in the sequence

$$S_1(p,q), S_2(p,q), \ldots, S_m(p,q)$$

and a (single vector) basis of the approxi-null space. A GCD finder constructed in this way can be illustrated in the following process.

First, we form $S_1(p,q)$, set the first permutation $P_1 = I_{(n-m+2)\times(n-m+2)}$, and calculate its QR decomposition

$$T_1 = S_1(p,q)P_1$$

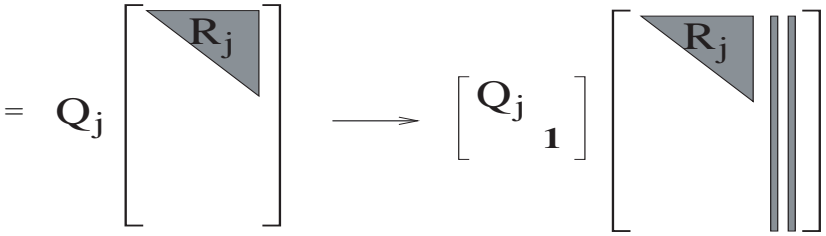


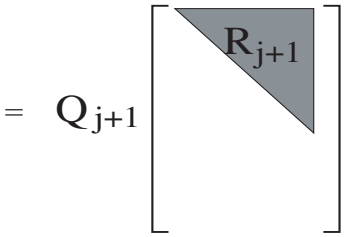
If $S_1(p,q)$ is approxi-rank deficient, then GCD $(p,q) = q$. The process needs to continue only if $S_1(p,q)$ is of full approxi-rank. In general, if $S_j(p,q)$ is of full approxi-rank with its pivoted QR decomposition $T_j = S_j(p,q)P_j = Q_jR_j$ being available, we attach one zero row to the bottom of T_j and add two columns

$$\begin{bmatrix} Q_j^\top & \\ & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ q_0 \\ \vdots \\ q_m \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} Q_j^\top & \\ & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ p_0 \\ \vdots \\ p_n \end{bmatrix}$$

to the right of the resulting matrix to form T_{j+1}. With a proper permutation matrix P_{j+1}, we have $T_{j+1}P_{j+1}^\top = S_{j+1}(p,q)$. Therefore,

$$T_j = S_j(p,q)P_j$$





$$= S_{j+1}(p,q)P_{j+1} = T_{j+1}.$$

Updating the QR decomposition of $T_{j+1} = S_{j+1}(p,q)P_{j+1}$ requires only $O(n+m)$ additional flops. We apply the iteration (3.2) on R_{j+1} for an approxi-null vector. If R_{j+1} (or, equivalently, $S_{j+1}(p,q)$) remains in full approxi-rank, the process continues to $j+2$ in a similar way. It stops at the (column permuted) kth Sylvester resultant matrix $T_k = S_k(p,q)P_k$, the first to be approxi-rank deficient.

It can be shown that the null space of T_k is of dimension one with a single null vector $\mathbf{z} \in \mathbb{R}^{n-m+2k}$ in its basis. Let

$$\begin{pmatrix} \mathbf{w} \\ -\mathbf{v} \end{pmatrix} = P_k^\top \mathbf{z} \quad \text{with } \mathbf{w} \in \mathbb{R}^k \text{ and } \mathbf{v} \in \mathbb{R}^{n-m+k}.$$

Then $\mathbf{v}$ and $\mathbf{w}$ are coefficient vectors of $v(x)$ and $w(x)$ satisfying (9.1). Now $u(x) = \mathrm{GCD}\,(p,q)$ is the quotient of $p(x)$ and $v(x)$. However, it is numerically unstable to use polynomial synthetic division $p(x) \div v(x)$ for finding $u(x)$ [19]. Instead, we use the "least squares division" [19] which solves the coefficient vector $\mathbf{u}$ of $u(x)$ as a least squares solution to

$$\begin{pmatrix} v_0 & & \\ v_1 & \ddots & \\ \vdots & \ddots & v_0 \\ v_s & & v_1 \\ & \ddots & \vdots \\ & & v_s \end{pmatrix} \mathbf{u} = \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ \\ \vdots \\ \vdots \\ p_n \end{pmatrix}, \quad \mathbf{u} \in \mathbb{R}^{m-k+2},\ s = n-m+k-1. \tag{9.2}$$

The procedure listed in Figure 9.1 illustrates the calculation of $\deg(\mathrm{GCD}\,(p,q))$ and the coefficients of $u(x)$, $v(x)$, and $w(x)$ in (9.1). To achieve highest attainable accuracy, the Gauss–Newton iteration on a quadratic system based on (9.1) can be applied to refine the GCD [19].

9.2. Nonisolated solutions to a polynomial system. When a numerical solution $\mathbf{x}_0$ of a system of polynomial equations

$$P(\mathbf{x}) = \big(p_1(\mathbf{x}), \ldots, p_n(\mathbf{x})\big) = 0, \quad \text{where } \mathbf{x} = (x_1, \ldots, x_n)^\top \in \mathbb{C}^n,$$

is obtained, we wish to identify whether $\mathbf{x}_0$ is an isolated solution of $P(\mathbf{x}) = 0$. While in the previous sections we mainly focused our attention on the development of our method and algorithm in the real vector space $\mathbb{R}^n$, the entire content remains valid in $\mathbb{C}^n$ with proper adjustments.

If the Jacobian of $P(\mathbf{x})$, denoted by $P_x(\mathbf{x})$, at $\mathbf{x}_0$ allows no small (relative to $\|P_x(\mathbf{x}_0)\|_\infty$) singular values, $\mathbf{x}_0$ is of course an isolated solution. When our rank-revealing algorithm is applied to $P_x(\mathbf{x}_0)$ and the result shows it admits very small singular values, $\mathbf{x}_0$ may lie on a solution component of $P(\mathbf{x}) = 0$ with positive dimension or it may still be an isolated zero with multiplicity ≥ 2. Our strategy to distinguish those cases is given below.

If $P_x(\mathbf{x}_0)$ permits only one singular value that appears tiny and if $\mathbf{x}_0$ is not an isolated solution, then $\mathbf{x}_0$ must lie on a one- (complex) dimensional solution component M of $P(\mathbf{x}) = 0$. We will begin to identify this path to a substantial length by a path following scheme developed in [9]. If this attempt fails, no such solution component M may exist and $\mathbf{x}_0$ will be classified as an isolated solution of $P(\mathbf{x}) = 0$.

```
Pseudocode GCD:
 input:  coefficient vectors for p(x), q(x)
 output:  d = deg(GCD (p, q)),
             coefficients of v(x) and w(x) in (9.1)
    QR decomposition QR = S_1(p, q)
    For j = 1, 2, ..., m do
        Gauss--Newton iteration (3.2) on R, get ϱ and x
        if ϱ is small enough, then
            extract coefficients of v(x) and w(x) from x
            solve (9.2) for the coefficients of u(x)
            exit
        else
            if j ≤ m then
                update Q_{j+1}R_{j+1} = S_{j+1}(p, q)P_{j+1}
            else
                deg(GCD (p, q)) = 0, v(x) = p(x), w(x) = q(x)
            end if
        end if
    end do
```

FIG. 9.1. *Pseudocode of GCD.*

When $P_x(\mathbf{x}_0)$ has $k > 1$ very small singular values as a result of our rank-revealing algorithm, we augment $P(\mathbf{x}) = 0$ with $k-1$ generic hyperplanes

$$\mathbf{a}_j^H(\mathbf{x} - \mathbf{x}_0) = 0, \quad j = 1, \ldots, k-1,$$

at $\mathbf{x}_0$. The enlarged system

$$\widehat{P}(\mathbf{x}) = \begin{cases} P(\mathbf{x}) = 0, \\ \mathbf{a}_1^H(\mathbf{x} - \mathbf{x}_0) = 0 \\ \vdots \\ \mathbf{a}_{k-1}^H(\mathbf{x} - \mathbf{x}_0) = 0 \end{cases} \tag{9.3}$$

will produce a one-dimensional component $\widehat{M}$ of $\widehat{P}(\mathbf{x}) = 0$ if the solution component M of $P(\mathbf{x}) = 0$ to which $\mathbf{x}_0$ belongs is of dimension k. Thus, the assertion that $\dim(M) = k$ is valid only if we can identify $\widehat{M}$ by following this path to a satisfactory length. If the path following cannot be carried out successfully, such a component $\widehat{M}$ may not exist. We will then remove hyperplane $\mathbf{a}_{k-1}^H(\mathbf{x} - \mathbf{x}_0) = 0$ in (9.3) and restart our effort to identify the one-dimensional component $\widehat{\widehat{M}}$ produced by

$$\widehat{\widehat{P}}(\mathbf{x}) = \begin{cases} P(\mathbf{x}) = 0, \\ \mathbf{a}_1^H(\mathbf{x} - \mathbf{x}_0) = 0 \\ \vdots \\ \mathbf{a}_{k-2}^H(\mathbf{x} - \mathbf{x}_0) = 0. \end{cases} \tag{9.4}$$

The existence of such a component $\widehat{\widehat{M}}$ of $\widehat{\widehat{P}} = 0$ implies the solution component M of $P(\mathbf{x}) = 0$ containing $\mathbf{x}_0$ is of dimension $k-1$. If it fails, the process may be continued in the same manner and the dimension of M will ultimately (very soon in practice) be determined. Of course, when $\dim(M) = 0$, $\mathbf{x}_0$ is an isolated zero even though $P_x(\mathbf{x}_0)$ may have very small singular values from our rank-revealing algorithm.

Example (see [15]). Consider the polynomial system $P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), p_3(\mathbf{x}))$, $\mathbf{x} = (u, v, w) \in \mathbb{C}^3$, where

$$\begin{aligned} p_1(\mathbf{x}) &= (v - u^2) \quad \cdot \quad (u^2 + v^2 + w^2 - 1)(u - 0.5), \\ p_2(\mathbf{x}) &= (w - u^3)(u^2 + v^2 + w^2 - 1)(v - 0.5), \\ p_3(\mathbf{x}) &= (v - u^2)(w - u^3)(u^2 + v^2 + w^2 - 1)(w - 0.5). \end{aligned}$$

Obviously, the solution set of $P(\mathbf{x}) = 0$ consists of

1. a two-dimensional component $u^2 + v^2 + w^2 = 1$;
2. four one-dimensional components
   (a) line $u = 0.5$, $v = (0.5)^3$;
   (b) line $u = \sqrt{0.5}$, $v = 0.5$;
   (c) line $u = -\sqrt{0.5}$, $v = 0.5$;
   (d) twisted cubic $v = u^2$, $w = u^3$;
3. one isolated solution $(u, v, w) = (0.5, 0.5, 0.5)$.

When the polyhedral homotopy continuation method [10] was used to solve $P(\mathbf{x}) = 0$, 129 numerical solutions were obtained. We applied our method to all those solutions, and the result shows

- 112 of them lie on the two-dimensional component,
- 16 of them lie on one-dimensional components (four on line 2a, four on line 2b, four on line 2c, four on line 2d),
- one isolated solution.

When we classified a solution $\mathbf{x}_0$ that is lying on a two-dimensional component of $P(\mathbf{x}) = 0$, for instance, we substituted $\mathbf{x}_0$ into $u^2 + v^2 + w^2 = 1$ to verify the accuracy of our identification, and the results were all accurate.

Acknowledgments. The authors wish to thank R. D. Fierro, P. C. Hansen, and P. S. K. Hansen for making UTV Tools freely available. In particular, the second author is grateful to P. C. Hansen for his helpful e-correspondence.

REFERENCES

[1] M. W. Berry, *Large scale sparse singular value computation*, Internat. J. Supercomput. Appl., 6 (1992), pp. 13–49.
[2] 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.
[3] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
[4] T. R. Chan, *Rank revealing QR factorizations*, Linear Algebra Appl., 88/89 (1987), pp. 67–82.
[5] F. Deprettere, *SVD and Signal Processing, Algorithms, Applications, and Architectures*, North–Holland, Amsterdam, 1988.
[6] R. D. Fierro, P. C. Hansen, and P. S. K. Hansen, *UTV tools: MATLAB templates for rank-revealing UTV decompositions*, Numer. Algorithms, 20 (1999), pp. 165–194.
[7] G. H. Golub, V. Klema, and G. W. Stewart, *Rank Degeneracy and Least Squares Problems*, Tech. rep. TR 456, University of Maryland, Baltimore, MD, 1976.
[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[9] Y. C. Kuo and T. Y. Li, *Determining Whether a Zero of a Polynomial System is Isolated*, preprint, 2003.

[10] T. Y. Li, *Numerical solution of multivariate polynomial systems by homotopy continuation methods*, in Acta Numerica, Acta Numer. 6, Cambridge University Press, Cambridge, UK, 199, pp. 399–436.
[11] R. Mathias and G. W. Stewart, *A block QR algorithm and the singular value decomposition*, Linear Algebra Appl., 182 (1993), pp. 91–100.
[12] L. Mirsky, *Symmetric gauge functions and unitarily invariant norms*, Quart. J. Math. Oxford Ser. (2), 11 (1960), pp. 50–59.
[13] M. Moonen and B. De Moor, *SVD and Signal Processing,* III*, Algorithms, Applications, and Architectures*, Elsevier, Amsterdam, 1995.
[14] D. Rupprecht, *An algorithm for computing certified approximate GCD of n univariate polynomials*, J. Pure Appl. Algebra, 139 (1999), pp. 255–284.
[15] A. J. Sommese, J. Verschelde, and C. W. Wampler, *Numerical decomposition of the solution sets of polynomial systems into irreducible components*, SIAM J. Numer. Anal., 38 (2001), pp. 2022–2046.
[16] G. W. Stewart, *UTV decompositions*, in Numerical Analysis 1993, D. F. Griffith and G. A. Watson, eds., Pitman Res. Notes Math. Ser. 303, Longman, Harlow, UK 1994, pp. 225–236.
[17] G. W. Stewart, *Matrix Algorithms: Basic Decompositions*, SIAM, Philadelphia, 1998.
[18] R. Vaccaro, *SVD and Signal Processing,* II*, Algorithms, Applications, and Architectures*, Elsevier, Amsterdam, 1991.
[19] Z. Zeng, *Computing multiple roots of inexact polynomials*, Math. Comp., 74 (2005), pp. 869–903.
[20] Z. Zeng, *Algorithm* 835: *MultRoot—a MATLAB package for computing polynomial roots and multiplicities*, ACM Trans. Math. Software, 30 (2004), pp. 218–236.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 4, pp. 947–961

PERTURBATION BOUNDS FOR ISOTROPIC INVARIANT SUBSPACES OF SKEW-HAMILTONIAN MATRICES*

DANIEL KRESSNER†

Abstract. We investigate the behavior of isotropic invariant subspaces of skew-Hamiltonian matrices under structured perturbations. It is shown that finding a nearby subspace is equivalent to solving a certain quadratic matrix equation. This connection is used to derive meaningful error bounds and condition numbers that can be used to judge the quality of invariant subspaces computed by strongly backward stable eigensolvers.

Key words. skew-Hamiltonian, invariant subspace, perturbation analysis, Sylvester equation, Riccati equation

AMS subject classifications. 47A15, 47A55, 65F15

DOI. 10.1137/S0895479803429752

1. Introduction.

A real $2n \times 2n$ matrix of the form

$$W = \begin{bmatrix} A & G \\ H & A^T \end{bmatrix}, \quad G = -G^T, \quad H = -H^T, \tag{1.1}$$

with $A, G, H \in \mathbb{R}^{n\times n}$ is called *skew-Hamiltonian*. The imposed structure has a number of consequences for the eigenvalues and eigenvectors of W; one is that each eigenvalue appears at least twice. Hence, well-known results from matrix perturbation theory predict that the eigenvectors of W are extremely ill conditioned; i.e., they may change drastically under small perturbations. For example, consider the parameter-dependent matrix

$$W(\varepsilon_1, \varepsilon_2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ \varepsilon_1 & \varepsilon_2 & 1 & 0 \\ -\varepsilon_2 & 0 & 0 & 2 \end{bmatrix}.$$

The vector $e_1 = [1, 0, 0, 0]^T$ is an eigenvector of $W(0, 0)$ associated with the eigenvalue $\lambda = 1$. No matter how small $\varepsilon_1 > 0$ is, any eigenvector of $W(\varepsilon_1, 0)$ associated with λ has the completely different form $[0, 0, \alpha, 0]^T$ for some $\alpha \neq 0$. On the other hand, $W(0, \varepsilon_2)$ has an eigenvector $[1, 0, 0, \varepsilon_2]^T$ rather close to e_1. The fundamental difference between $W(\varepsilon_1, 0)$ and $W(0, \varepsilon_2)$ is that the latter is a skew-Hamiltonian matrix while the former is not.

In this paper we investigate the behavior of eigenvectors of skew-Hamiltonian matrices under perturbations that are structure-preserving, as in the case of $W(0, \varepsilon_2)$. More generally, the discussion is concerned with isotropic invariant subspaces, which are, loosely speaking, the invariant subspaces of W associated with at most one copy of each eigenvalue. We derive error bounds that allow users of strongly backward stable

*Received by the editors June 7, 2003; accepted for publication (in revised form) by B. T. Kågström March 3, 2004; published electronically May 6, 2005. This research was supported by the DFG Research Center "Mathematics for Key Technologies" (FZT 86) in Berlin and a Marie Curie Fellowship in the frame of the Control Training Site (MCFI-2001-00403).

http://www.siam.org/journals/simax/26-4/42975.html

†Institut für Mathematik, MA 4-5, TU Berlin, Str. des 17. Juni 136, D-10623 Berlin, Germany (kressner@math.tu-berlin.de).

eigensolvers [11, 19] to quantify their obtained results. Furthermore, applications that directly depend on the computation of isotropic invariant subspaces such as certain Riccati equations [13] and quadratic eigenvalue problems [18] may benefit from these bounds.

The perturbation theory given here owes much to the fact that there exist considerably simple condensed forms for skew-Hamiltonian matrices. Section 2 reviews some of these forms along with other theoretical tools required later on. In section 3, the connection between finding a nearby isotropic subspace and solving a quadratic matrix equation is explained. The solution of this equation is complicated by an artificial singularity; its lengthy derivation is described in section 4. The subsequent section contains the central result of this work: Theorem 5.2 gives an upper bound for the sensitivity of isotropic invariant subspaces. This will lead us to define a corresponding condition number, and section 6 contains some discussion on how this quantity can be computed. Finally, in section 7 a numerical example is presented to illustrate the use of the derived condition number.

2. Basic tools. Equivalent to the block representation (1.1), a skew-Hamiltonian matrix W is characterized by the fact that J_nW is skew-symmetric, where $J_n = \left[\begin{smallmatrix} 0 & I_n \\ -I_n & 0 \end{smallmatrix}\right]$ and I_n is the $n \times n$ identity matrix. In the following we will drop the subscript n whenever the dimension of the corresponding matrix is clear from its context. A matrix $S \in \mathbb{R}^{2n\times 2n}$ is called *symplectic* if $S^TJS = J$. It is easy to show that in this case $JS^{-1}WS$ is skew-symmetric; thus symplectic similarity transformations preserve skew-Hamiltonian structures. Moreover, an orthogonal matrix U is symplectic if and only if it has the representation

$$U = \begin{bmatrix} U_1 & U_2 \\ -U_2 & U_1 \end{bmatrix}, \quad U_1, U_2 \in \mathbb{R}^{n\times n}. \tag{2.1}$$

We will call such a matrix *orthogonal symplectic*. An important property of U is that its first $k \leq n$ columns span an isotropic subspace.

DEFINITION 2.1. *A subspace $\mathcal{X} \subset \mathbb{R}^{2n}$ is called* isotropic *if $J\mathcal{X} \perp \mathcal{X}$.*

Van Loan [19] showed that for any skew-Hamiltonian matrix W there exists an orthogonal symplectic matrix U so that

$$U^TWU = \begin{bmatrix} \tilde{A} & \tilde{G} \\ 0 & \tilde{A}^T \end{bmatrix}, \tag{2.2}$$

where $\tilde{A}$ is in real Schur form. Moreover, real eigenvalues and complex conjugate pairs of eigenvalues may appear in any desirable order on the diagonal of $\tilde{A}$. Closely related to (2.2) is the following characterization of isotropic invariant subspaces of W.

LEMMA 2.2. *Let $W \in \mathbb{R}^{2n\times 2n}$ be a skew-Hamiltonian matrix, and let $X \in \mathbb{R}^{2n\times k}$ $(k \leq n)$ have orthonormal columns. Then the columns of X span an isotropic invariant subspace of W if and only if there exists an orthogonal symplectic matrix $U = [X, Z, J^TX, J^TZ]$ with some $Z \in \mathbb{R}^{2n\times(n-k)}$ so that*

$$U^TWU = \begin{array}{c} \\ k \\ n-k \\ k \\ n-k \end{array} \begin{array}{c} \begin{array}{cccc} k & n-k & k & n-k \end{array} \\ \begin{bmatrix} A_{11} & A_{12} & G_{11} & G_{12} \\ 0 & A_{22} & -G_{12}^T & G_{22} \\ 0 & 0 & A_{11}^T & 0 \\ 0 & H_{22} & A_{12}^T & A_{22}^T \end{bmatrix} \end{array}. \tag{2.3}$$

Proof. Assume that the columns of X span an isotropic subspace. Then the symplectic QR factorization [2] can be used to construct an orthogonal symplectic matrix $U = [X, Z, J^T X, J^T Z]$. Moreover, if the columns of X span an invariant subspace, then $[Z, J^T X, J^T Z]^T W X = 0$, completing the proof of (2.3). The other direction is straightforward. ∎

As the spectral properties of $A_{11} = X^T W X$ and

$$\begin{bmatrix} A_{22} & G_{22} \\ H_{22} & A_{22}^T \end{bmatrix} = [Z, J^T Z]^T W [Z, J^T Z]$$

do not depend on the choice of bases, the following definition can be used to adapt the notion of simple invariant subspaces to skew-Hamiltonian matrices.

DEFINITION 2.3. *Let the orthonormal columns of $X \in \mathbb{R}^{2n\times k}$ span an isotropic invariant subspace $\mathcal{X}$ of a skew-Hamiltonian matrix W. Furthermore, choose $Z \in \mathbb{R}^{2n\times(n-k)}$ so that $U = [X, Z, J^T X, J^T Z]$ is orthogonal symplectic and $U^T W U$ has the form* (2.3). *Then $\mathcal{X}$ is called* semisimple *if $\lambda(A_{11}) \cap \lambda(\left[\begin{smallmatrix} A_{22} & G_{22} \\ H_{22} & A_{22}^T \end{smallmatrix}\right]) = \emptyset$ and A_{11} is nonderogatory, i.e., each eigenvalue of A_{11} has geometric multiplicity one.*

Semisimple subspaces allow us to block diagonalize W by a simple transformation. For this purpose, we require two facts about Sylvester equations. The first is a well-known result. Proofs can be found in many places; see, e.g., [8].

PROPOSITION 2.4. *The Sylvester equation*

$$AP - PB = C$$

with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times m}$, and $C \in \mathbb{R}^{n\times m}$ has a unique solution $P \in \mathbb{R}^{n\times m}$ if and only if $\lambda(A) \cap \lambda(B) = \emptyset$.

The second is concerned with a certain type of singular Sylvester equations that do not fit into the framework of Proposition 2.4.

PROPOSITION 2.5. *The Sylvester equation*

$$AP - PA^T = G \tag{2.4}$$

is solvable for all skew-symmetric matrices G if and only if A is nonderogatory. In this case, any solution P to (2.4) *is real and symmetric.*

Proof. This result can be found in [4]. Actually, the second part is not explicitly stated there but follows easily from the proof of Proposition 5 in [4]. ∎

Propositions 2.4 and 2.5 can be combined to successively annihilate the blocks A_{12}, G_{12}, and G_{11} in the block representation (2.3) for a semisimple subspace. To see this, solve

$$A_{11} \begin{bmatrix} P_1 & P_2 \end{bmatrix} - \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} A_{22} & G_{22} \\ H_{22} & A_{22}^T \end{bmatrix} = - \begin{bmatrix} A_{12} & G_{12} \end{bmatrix},$$

and construct the symplectic matrix

$$S_P = \begin{bmatrix} I & P_1 & -P_1 P_2^T & P_2 \\ 0 & I & P_2^T & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & -P_1^T & I \end{bmatrix},$$

yielding

$$S_P^{-1} \begin{bmatrix} A_{11} & A_{12} & G_{11} & G_{12} \\ 0 & A_{22} & -G_{12}^T & G_{22} \\ 0 & 0 & A_{11}^T & 0 \\ 0 & H_{22} & A_{12}^T & A_{22}^T \end{bmatrix} S_P = \begin{bmatrix} A_{11} & 0 & \tilde{G}_{11} & 0 \\ 0 & A_{22} & 0 & G_{22} \\ 0 & 0 & A_{11}^T & 0 \\ 0 & H_{22} & 0 & A_{22}^T \end{bmatrix} \tag{2.5}$$

with a skew-symmetric matrix $\tilde{G}_{11}$. Next, use a solution of

$$A_{11}Q - QA_{11}^T = -\tilde{G}_{11} \tag{2.6}$$

to construct

$$S_Q = \begin{bmatrix} I & 0 & Q & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}.$$

The matrix S_Q is symplectic since Proposition 2.5 guarantees that Q is symmetric. If the similarity transformation associated with S_Q is applied to the right-hand side of (2.5), then the block $\tilde{G}_{11}$ is annihilated. Note that there is a lot of freedom in the choice of Q as (2.6) admits infinitely many solutions. From a numerical point of view the matrix Q should be chosen so that the condition number of the product $S_P S_Q$ is as small as possible.

3. Perturbations and a quadratic matrix equation. Consider an isotropic invariant subspace $\mathcal{X}$ of a skew-Hamiltonian matrix W. Given a skew-Hamiltonian perturbation E of small norm we now investigate the question of whether $W + E$ has an isotropic invariant subspace $\hat{\mathcal{X}}$ close to $\mathcal{X}$. What follows is in many aspects similar to the treatment of general matrices by Stewart [14, 15]; however, we end up with a quadratic matrix equation of quite a different nature.

Let the columns of X form an orthonormal basis for $\mathcal{X}$. Apply Lemma 2.2 to construct a matrix $Y = [Z, J^T Z, J^T X]$ so that $\tilde{U} = [X, Y]$ is an orthogonal matrix. Note that $\tilde{U}^T(W + E)\tilde{U}$ is a permuted skew-Hamiltonian matrix and can be partitioned as

$$\tilde{U}^T(W+E)\tilde{U} = \begin{array}{c} \\ k \\ 2(n-k) \\ k \end{array} \begin{array}{c} \begin{array}{ccc} k & 2(n-k) & k \end{array} \\ \begin{bmatrix} W_{11} & W_{23}^T J_{n-k}^T & W_{13} \\ W_{21} & W_{22} & W_{23} \\ W_{31} & W_{21}^T J_{n-k} & W_{11}^T \end{bmatrix} \end{array}, \tag{3.1}$$

where W_{13} and W_{31} are skew-symmetric matrices, and W_{22} is skew-Hamiltonian. For $E = 0$, the matrices W_{21} and W_{31} are zero and the other blocks in (3.1) correspond to the block representation (2.3) as follows:

$$W_{11} = A_{11},\ W_{13} = G_{11},\ W_{22} = \begin{bmatrix} A_{22} & G_{22} \\ H_{22} & A_{22}^T \end{bmatrix},\ W_{23} = \begin{bmatrix} -G_{12}^T \\ A_{12}^T \end{bmatrix}.$$

Now, let

$$\hat{X} = \left(X + Y \begin{bmatrix} P \\ Q \end{bmatrix} \right) (I + P^T P + Q^T Q)^{-1/2}, \tag{3.2}$$

$$\hat{Y} = (Y - X \begin{bmatrix} P^T & Q^T \end{bmatrix}) \left(I + \begin{bmatrix} P \\ Q \end{bmatrix} \begin{bmatrix} P^T & Q^T \end{bmatrix} \right)^{-1/2}, \tag{3.3}$$

where $P \in \mathbb{R}^{2(n-k)\times k}$ and $Q \in \mathbb{R}^{k\times k}$ are matrices to be determined so that $\hat{\mathcal{X}} = \operatorname{span}(\hat{X})$ is an isotropic invariant subspace of $W + E$. This is equivalent to the conditions $Q^T - Q = P^T J P$ and $\hat{Y}^T(W + E)\hat{X} = 0$. In terms of (3.1), the latter can be written as

$$\begin{bmatrix} P \\ Q \end{bmatrix} W_{11} - \begin{bmatrix} W_{22} & W_{23} \\ W_{21}^T J & W_{11}^T \end{bmatrix} \begin{bmatrix} P \\ Q \end{bmatrix} + \begin{bmatrix} P \\ Q \end{bmatrix} \begin{bmatrix} JW_{23} \\ W_{13}^T \end{bmatrix}^T \begin{bmatrix} P \\ Q \end{bmatrix} = \begin{bmatrix} W_{21} \\ W_{31} \end{bmatrix}. \tag{3.4}$$

Once we have solved (3.4), the sines of the canonical angles between $\mathcal{X}$ and $\hat{\mathcal{X}}$ are the singular values of

$$Y^T\hat{X} = \begin{bmatrix} P \\ Q \end{bmatrix} (I + P^TP + Q^TQ)^{-1/2};$$

see, e.g., [16, sect. I.5]. We will see that (3.4) may admit infinitely many solutions satisfying $Q^T - Q = P^TJP$. In the interest of a small distance between $\mathcal{X}$ and $\hat{\mathcal{X}}$, a solution of small norm is preferred.

4. A solution of the quadratic matrix equation. Solving (3.4) is complicated by two facts. First, we have to guarantee that the solution satisfies $Q^T - Q = P^TJP$, and second, the linear part of (3.4) is close to a singular linear matrix equation if $W_{21} \approx 0$. Unfortunately, it is not easy to see from the present formulation of (3.4) that this singularity is, due to the special structure of the nonlinearities and the right-hand side, artificial. Both issues can be more easily addressed after a reformulation of (3.4).

4.1. Skew-symmetrizing the bottom part. Let

$$R = Q + P^T\tilde{J}P, \quad \tilde{J} = \begin{bmatrix} 0 & I_{n-k} \\ 0 & 0 \end{bmatrix}; \tag{4.1}$$

then R is symmetric if and only if $Q^T - Q = P^TJP$. The following lemma reveals a particular useful nonlinear matrix equation satisfied by (P, R).

LEMMA 4.1. *Let $R = Q + P^T\tilde{J}P$ be symmetric. Then the matrix pair (P, Q) is a solution of* (3.4) *if and only if (P, R) is a solution of*

$$\begin{bmatrix} P \\ R \end{bmatrix} W_{11} - \begin{bmatrix} W_{22} & W_{23} \\ W_{21}^TJ & W_{11}^T \end{bmatrix} \begin{bmatrix} P \\ R \end{bmatrix} + \begin{bmatrix} \Phi_1(P,R) \\ \Phi_2(P,R) - P^TJW_{21} \end{bmatrix} = \begin{bmatrix} W_{21} \\ W_{31} \end{bmatrix}, \tag{4.2}$$

where

$$\begin{aligned} \Phi_1(P,R) &= W_{23}(P^T\tilde{J}P) + P(JW_{23})^TP + PW_{13}(R - P^T\tilde{J}P), \\ \Phi_2(P,R) &= (R - P^T\tilde{J}^TP)W_{23}^TJ^TP - P^TJW_{23}(R - P^T\tilde{J}^TP)^T \\ &\quad + (R - P^T\tilde{J}P)^TW_{13}(R - P^T\tilde{J}P) - P^TJW_{22}P. \end{aligned}$$

Proof. Adding the top part of (3.4) premultiplied by P^TJ,

$$P^TJW_{21} = P^TJPW_{11} - P^TJW_{22}P - P^TJW_{23}Q + P^TJP(JW_{23})^TP + P^TJPW_{13}Q,$$

to the bottom part of (3.4) yields the transformed equation (4.2) after some basic algebraic manipulations. ∎

The reformulated equation (4.2) has the advantage that the nonlinear function $\Phi_2(P, R)$, the right-hand side term W_{31}, as well as the coupling term $-W_{21}^TJP - P^TJW_{21}$ are skew-symmetric. Hence, these terms belong to the range of the operator $R \mapsto RW_{11} - W_{11}^TR$ provided that W_{11} is nonderogatory. This indicates that the singularity caused by this operator is indeed artificial.

4.2. Solving the decoupled linearized equation. Linearizing (4.2) around $(P, R) = (0, 0)$ yields

$$\tilde{\mathcal{T}}(P,R) = \begin{bmatrix} W_{21} \\ W_{31} \end{bmatrix}, \tag{4.3}$$

where the operator $\tilde{\mathcal{T}} : \mathbb{R}^{2(n-k)\times k} \times \mathbb{R}^{k\times k} \to \mathbb{R}^{2(n-k)\times k} \times \mathbb{R}^{k\times k}$ is given by

$$\tilde{\mathcal{T}} : (P, R) \mapsto \begin{bmatrix} P \\ R \end{bmatrix} W_{11} - \begin{bmatrix} W_{22} & W_{23} \\ W_{21}^T J & W_{11}^T \end{bmatrix} \begin{bmatrix} P \\ R \end{bmatrix} - \begin{bmatrix} 0 \\ P^T J W_{21} \end{bmatrix}.$$

Note that we sometimes identify $(X, Y) \sim \left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right]$ for notational convenience. It is assumed that the perturbation E is considerably small implying that W_{21} is small. Hence, $W_{21}^T JP$ and $P^T J W_{21}$ can be regarded as weak coupling terms. Let us neglect these terms and consider the operator

$$\mathcal{T} : (P, R) \mapsto \begin{bmatrix} P \\ R \end{bmatrix} W_{11} - \begin{bmatrix} W_{22} & W_{23} \\ 0 & W_{11}^T \end{bmatrix} \begin{bmatrix} P \\ R \end{bmatrix}, \tag{4.4}$$

which allows for an easy characterization. In the following lemma, $\mathrm{Sym}(k)$ denotes the set of all symmetric $k \times k$ matrices, and $\mathrm{Skew}(k)$ denotes the set of all skew-symmetric $k \times k$ matrices.

LEMMA 4.2. *Consider the operator $\mathcal{T}$ defined by* (4.4) *with domain and codomain restricted to* $\operatorname{dom} \mathcal{T} = \mathbb{R}^{2(n-k)\times k} \times \mathrm{Sym}(k)$ *and* $\operatorname{codom} \mathcal{T} = \mathbb{R}^{2(n-k)\times k} \times \mathrm{Skew}(k)$, *respectively. Then $\mathcal{T}$ is onto if and only if W_{11} is nonderogatory and $\lambda(W_{11}) \cap \lambda(W_{22}) = \emptyset$.*

Proof. If W_{11} is nonderogatory and $\lambda(W_{11}) \cap \lambda(W_{22}) = \emptyset$, then we can apply Propositions 2.5 and 2.4 combined with backward substitution to show that $\mathcal{T}$ is onto. For the other direction, assume that $\mathcal{T}$ is onto. Proposition 2.5 implies that W_{11} is nonderogatory; it remains to show that $\lambda(W_{11}) \cap \lambda(W_{22}) = \emptyset$. By continuity, we may assume w.l.o.g. that there is a nonsingular matrix X so that $\Lambda = X^{-1} W_{11} X$ is diagonal with diagonal elements $\lambda_1, \ldots, \lambda_k \in \mathbb{C}$. Then there is a matrix $\tilde{R}_0 \in \mathbb{C}^{k\times k}$ so that every solution of the transformed equation $\tilde{R}\Lambda - \bar{\Lambda}\tilde{R} = X^{-1} W_{31} X$ has the form

$$\tilde{R} = \tilde{R}_0 + \sum_{i=1}^{k} \alpha_i e_i e_i^T, \quad \alpha_1, \ldots, \alpha_k \in \mathbb{C}.$$

Inserting this representation into the equation $\tilde{P}\Lambda - W_{22}\tilde{P} - W_{23} X^{-T} \tilde{R} = W_{13} X$ leads to the k separate equations

$$\begin{bmatrix} \lambda_i I - W_{22} & b_i \end{bmatrix} \begin{bmatrix} \tilde{p}_i \\ \alpha_i \end{bmatrix} = (W_{13} X + W_{23} \tilde{R}_0) e_i, \tag{4.5}$$

where $\tilde{p}_i$ and b_i denote the ith columns of $\tilde{P}$ and $W_{23} X^{-T}$, respectively. Equation (4.5) has a solution for any $W_{13} \in \mathbb{R}^{2(n-k)\times k}$ if and only if $[\lambda_i I - W_{22}, b_i]$ has full rank $2(n-k)$. This implies $\lambda_i \notin \lambda(W_{22})$, since otherwise

$$\operatorname{rank}\big(\begin{bmatrix} \lambda_i I - W_{22} & b_i \end{bmatrix}\big) \leq \operatorname{rank}(\lambda_i I - W_{22}) + 1 \leq 2(n-k) - 1,$$

where we used the fact that the geometric multiplicity of each eigenvalue of the skew-Hamiltonian matrix W_{22} is at least two [4, Thm. 1]. Thus $\lambda(W_{11}) \cap \lambda(W_{22}) = \emptyset$, which concludes the proof. ∎

For the remainder of this section only the restricted operator $\mathcal{T}$ will be considered, and it will be assumed that this operator is onto. Note that for $E = 0$ the latter is equivalent to the assumption that $\mathcal{X}$ is semisimple; see Definition 2.3. The dimensions of the matrix spaces $\mathrm{Skew}(k)$ and $\mathrm{Sym}(k)$ differ by k. More precisely, it can be shown that the set of solutions corresponding to a particular right-hand side in the codomain

of $\mathcal{T}$ form an affine subspace of dimension k [4]. In view of an earlier remark, one should pick a solution that has minimal norm. Using the Frobenius norm this solution is uniquely determined as the following lemma shows.

LEMMA 4.3. *Let $\mathcal{T}$ be defined as in* (4.4)*, and let $(W_{21}, W_{31}) \in \operatorname{codom} \mathcal{T}$. Then there is one and only one matrix pair $(P_\star, R_\star) \in \operatorname{dom} \mathcal{T}$ satisfying*

$$\|(P_\star, R_\star)\|_F = \min_{(P,R)\in \operatorname{dom}\, \mathcal{T}} \left\{ \|(P,R)\|_F \mid \mathcal{T}(P,R) = \begin{bmatrix} W_{21} \\ W_{31} \end{bmatrix} \right\}. \tag{4.6}$$

Proof. Using the second part of Proposition 2.5 the constraint $(P, R) \in \operatorname{dom} \mathcal{T}$ in (4.6) can be dropped. Let us define

$$K_{\mathcal{T}} := W_{11}^T \otimes I - I \otimes \begin{bmatrix} W_{22} & W_{23} \\ 0 & W_{11}^T \end{bmatrix},$$

where "$\otimes$" denotes the Kronecker product of two matrices [5, sect. 4.5.5]. Using the vec operator, which stacks the columns of a matrix into one long vector, the minimization problem (4.6) can be written in the form

$$\min_{x\in\mathbb{R}^{(2n-k)\times k}} \{\|x\|_2 \ : \ K_{\mathcal{T}} \cdot x = w\}, \tag{4.7}$$

where $w = \operatorname{vec}([\begin{smallmatrix} W_{21} \\ W_{31} \end{smallmatrix}])$. Well-known results about linear least-squares problems show that (4.7) has a unique minimum given by $K_{\mathcal{T}}^\dagger \cdot w$, where $K_{\mathcal{T}}^\dagger$ denotes the pseudo-inverse of $K_{\mathcal{T}}$ [5, sect. 5.5.4]. ∎

This lemma allows us to define an operator

$$\mathcal{T}^\dagger : \operatorname{codom} \mathcal{T} \to \operatorname{dom} \mathcal{T}$$

which maps a matrix pair (W_{21}, W_{31}) to the solution of (4.6). A sensible choice of norm for $\mathcal{T}^\dagger$ is the one induced by the Frobenius norm:

$$\|\mathcal{T}^\dagger\| := \sup_{\substack{\|(W_{21},W_{31})\|_F=1 \\ (W_{21},W_{31})\in\operatorname{codom}\, \mathcal{T}}} \|\mathcal{T}^\dagger(W_{21}, W_{31})\|_F. \tag{4.8}$$

4.3. Solving the coupled linearized equation. The key to solving the coupled equation (4.3) is to note that $\tilde{\mathcal{T}}$ can be decomposed into $\mathcal{T} - \triangle\mathcal{T}_W$, where $\triangle\mathcal{T} : \operatorname{dom} \mathcal{T} \to \operatorname{codom} \mathcal{T}$ is defined by

$$\triangle\mathcal{T} : (P, R) \mapsto \begin{bmatrix} 0 \\ P^T J W_{21} + W_{21}^T J P \end{bmatrix}. \tag{4.9}$$

This implies that the composed operator $\mathcal{T}^\dagger \circ \triangle\mathcal{T} : \operatorname{dom} \mathcal{T} \to \operatorname{dom} \mathcal{T}$ is well defined; its norm is again the one induced by the Frobenius norm.

LEMMA 4.4. *If $\mathcal{T}$ is onto and $\delta := \|\mathcal{T}^\dagger \circ \triangle\mathcal{T}\| < 1$, then*

$$\tilde{\mathcal{T}}^\dagger(W_{21}, W_{31}) := \sum_{i=0}^{\infty} (\mathcal{T}^\dagger \circ \triangle\mathcal{T})^i \circ \mathcal{T}^\dagger(W_{21}, W_{31}) \tag{4.10}$$

is a solution of (4.3).

Proof. If $\delta < 1$, then

$$\left\| \sum_{i=0}^{\infty} (\mathcal{T}^\dagger \circ \triangle\mathcal{T})^i \circ \mathcal{T}^\dagger \right\| \leq \sum_{i=0}^{\infty} \delta^i \|\mathcal{T}^\dagger\| = \frac{\|\mathcal{T}^\dagger\|}{1-\delta}, \tag{4.11}$$

implying that the infinite sum in (4.10) converges absolutely. Moreover, premultiplying (4.10) with $\mathcal{T} - \triangle\mathcal{T}$ shows that $\tilde{\mathcal{T}}^{\tilde{\dagger}}(W_{21}, W_{31})$ solves (4.3). ∎

Inequality (4.11) yields the bound

$$\|\tilde{\mathcal{T}}^{\tilde{\dagger}}(W_{21}, W_{31})\|_F \leq \frac{\|\mathcal{T}^\dagger\|}{1-\delta} \cdot \|(W_{21}, W_{31})\|_F. \tag{4.12}$$

An upper bound for the quantity δ is clearly given by $2\|\mathcal{T}^\dagger\|\|W_{21}\|_F$. It should be stressed that $\tilde{\mathcal{T}}^{\tilde{\dagger}} : \operatorname{codom} \mathcal{T} \to \operatorname{dom} \mathcal{T}$ does not necessarily give the solution of smallest norm. However, if $\|\triangle\mathcal{T}\|$ is sufficiently small, it can be expected to be rather close to it.

LEMMA 4.5. *Under the assumption of Lemma* 4.4*, let* $\tilde{\mathcal{T}}^\dagger : \operatorname{codom} \mathcal{T} \to \operatorname{dom} \mathcal{T}$ *denote the operator that maps a pair* (W_{21}, W_{31}) *to the minimal norm solution of the coupled equation* (4.3)*. Then*

$$\lim_{\triangle\mathcal{T}\to 0} \tilde{\mathcal{T}}^{\tilde{\dagger}} = \lim_{\triangle\mathcal{T}\to 0} \tilde{\mathcal{T}}^\dagger = \mathcal{T}^\dagger. \tag{4.13}$$

Proof. Lemma 4.4 shows that the coupled equation (4.3) has, for a given right-hand side in $\operatorname{codom} \mathcal{T}$, a nonempty set of solutions. This set is, according to Proposition 2.5, a subset of $\operatorname{dom} \mathcal{T}$. The solution of minimal norm is uniquely defined for reasons similar to those that have been used in the proof of Lemma 4.3. Hence, the operator $\tilde{\mathcal{T}}^\dagger$ is well defined. By checking the four Penrose conditions it can be shown that $\tilde{\mathcal{T}}^{\tilde{\dagger}} = (\mathcal{T} - \triangle\mathcal{T} \circ (\mathcal{T}^\dagger \circ \mathcal{T}))^\dagger$. Equalities (4.13) follow from the fact that the ranges of $\mathcal{T} - \triangle\mathcal{T} \circ (\mathcal{T}^\dagger \circ \mathcal{T}), \tilde{\mathcal{T}}$, and $\mathcal{T}$ have equal dimensions [16, sect. III.3]. ∎

We remark that Lemmas 4.4 and 4.5 are not restricted to perturbations of the form (4.9). In fact, they hold for any $\triangle\mathcal{T} : \operatorname{dom} \mathcal{T} \to \operatorname{codom} \mathcal{T}$ satisfying $\|\mathcal{T}^\dagger \circ \triangle\mathcal{T}\| < 1$.

4.4. Solving the nonlinear equation. Using the terminology developed above, we can rewrite the nonlinear equation (4.2) in the more convenient form

$$\tilde{\mathcal{T}}(P, R) + \Phi(P, R) = \begin{bmatrix} W_{21} \\ W_{31} \end{bmatrix}, \tag{4.14}$$

where $\Phi(P, R) = [\Phi_1(P, R)^T, \Phi_2(P, R)^T]^T$.

THEOREM 4.6. *Let the matrices* W_{ij} *be defined by* (3.1) *and assume that the operator* $\mathcal{T}$ *defined by* (4.4) *is onto in the sense of Lemma* 4.2*. Assume that* $\delta = 2\|\mathcal{T}^\dagger\|\|W_{21}\|_F < 1$*, where* $\|\mathcal{T}^\dagger\|$ *is defined by* (4.8)*. Set*

$$\gamma = \|(W_{21}, W_{31})\|_F, \quad \eta = \left\| \begin{bmatrix} W_{23}^T J^T & W_{13} \\ W_{22} & W_{23} \end{bmatrix} \right\|_F, \quad \kappa = \frac{\|\mathcal{T}^\dagger\|}{1-\delta}.$$

Then if

$$8\gamma\kappa < 1, \qquad 20\gamma\eta\kappa^2 < 1,$$

there is a solution (P, R) *of* (4.14) *satisfying*

$$\|(P, R)\|_F \leq 2\gamma\kappa. \tag{4.15}$$

Proof. We adapt the technique used by Stewart [15, sect. 3] and solve (4.14) by constructing an iteration. First, some facts about the nonlinearities are required:

$$\begin{aligned}
&\|\Phi_1(P,R)\|_F \le \|W_{13}\|_F(\|P\|_F\|R\|_F + \|P\|_F^3) + 2\|W_{23}\|_F\|P\|_F^2,\\
&\|\Phi_2(P,R)\|_F \le \eta\|(P,R)\|_F^2 + \|W_{13}\|_F(2\|P\|_F^2\|R\|_F + \|P\|_F^4) + 2\|W_{23}\|_F\|P\|_F^3,\\
\Rightarrow\ &\|\Phi(P,R)\|_F \le (1+\sqrt{3})\eta\|(P,R)\|_F^2 + (\sqrt{2}+\sqrt{3})\eta\|(P,R)\|_F^3 + \eta\|(P,R)\|_F^4.
\end{aligned}$$

Using a rough estimate, we have $\|\Phi(P,R)\|_F \le 4\eta\|(P,R)\|_F^2$ for $\|(P,R)\|_F \le 1/4$. Similarly, it can be shown that

$$\begin{aligned}
\|\Phi(\hat{P},\hat{R}) - \Phi(P,R)\|_F \le\ &[2(1+\sqrt{3})\eta\max\{\|(\hat{P},\hat{R})\|_F, \|(P,R)\|_F\}\\
&+4(\sqrt{2}+\sqrt{3})\eta\max\{\|(\hat{P},\hat{R})\|_F, \|(P,R)\|_F\}^2\\
&+8\eta\max\{\|(\hat{P},\hat{R})\|_F, \|(P,R)\|_F\}^3] \cdot \|(\hat{P}-P,\hat{R}-R)\|_F\\
\le\ &10\eta\max\{\|(\hat{P},\hat{R})\|_F, \|(P,R)\|_F\} \cdot \|(\hat{P}-P,\hat{R}-R)\|_F,
\end{aligned}$$

where the latter inequality holds for $\max\{\|(P,R)\|_F, \|(\hat{P},\hat{R})\|_F\} \le 1/4$. Next, we define a sequence by $(P_0,R_0) = (0,0)$ and

$$(P_{k+1},R_{k+1}) = \tilde{\mathcal{T}}^{\dagger}(W_{21},W_{31}) + \tilde{\mathcal{T}}^{\dagger} \circ \Phi(P_k,R_k).$$

Note that this iteration is well defined as $\Phi : \operatorname{dom}\mathcal{T} \to \operatorname{codom}\mathcal{T}$. We show by induction that the iterates stay bounded. Under the assumption $\|(P_k,R_k)\| < 2\gamma\kappa \le 1/4$, it follows that

$$\|(P_{k+1},R_{k+1})\|_F \le \kappa(\gamma + 4\eta\|(P_k,R_k)\|_F^2) < 2\gamma\kappa.$$

The operator $\tilde{\mathcal{T}}^{\dagger}\Phi$ is a contraction on $\mathcal{D} = \{(P,R) \ : \ \|(P,R)\|_F < 2\gamma\kappa\}$ since

$$\|\tilde{\mathcal{T}}^{\dagger} \circ \Phi(\hat{P},\hat{R}) - \tilde{\mathcal{T}}^{\dagger} \circ \Phi(P,R)\|_F \le 20\gamma\eta\kappa^2\|(\hat{P}-P,\hat{R}-R)\|_F < \|(\hat{P}-P,\hat{R}-R)\|_F$$

for all $(P,R) \in \mathcal{D}$ and $(\hat{P},\hat{R}) \in \mathcal{D}$. Thus, the contraction mapping theorem [12] shows that the sequence (P_k,R_k) converges to a fixed point, which solves (4.14). ∎

COROLLARY 4.7. *Under the assumptions of Theorem* 4.6*, there is a solution* (P,Q) *of the quadratic matrix equation* (3.4) *satisfying* $Q^T - Q = P^T J P$ *and*

$$\|(P,Q)\|_F \le 2\gamma\kappa + 4\gamma^2\kappa^2 < 2.5\gamma\kappa.$$

Proof. The result is a direct consequence of the relationship $Q = R - P^T\tilde{J}P$. ∎

5. Perturbation bounds and a condition number. From the discussion in section 3 it follows that Corollary 4.7 yields the existence of an isotropic invariant subspace $\hat{\mathcal{X}}$ of $W+E$ close to $\mathcal{X}$, which is an isotropic invariant subspace of the unperturbed matrix W.

COROLLARY 5.1. *Under the assumptions of Theorem* 4.6*, there is an isotropic invariant subspace* $\hat{\mathcal{X}}$ *of the skew-Hamiltonian matrix* $W+E$ *so that*

$$\sqrt{\tan^2\theta_1(\mathcal{X},\hat{\mathcal{X}}) + \cdots + \tan^2\theta_k(\mathcal{X},\hat{\mathcal{X}})} \le 2\gamma\kappa + 4\gamma^2\kappa^2 < 2.5\gamma\kappa, \tag{5.1}$$

where $\theta_i(\mathcal{X},\hat{\mathcal{X}})$, $i = 1,\ldots,k$, *are the canonical angles between* $\mathcal{X}$ *and* $\hat{\mathcal{X}}$.

Proof. Inequality (5.1) follows from Corollary 4.7 using the fact that $\tan\theta_i(\mathcal{X},\hat{\mathcal{X}})$, $i=1,\dots,k$, are the singular values of the matrix $[P^T,Q^T]^T$. ∎

The catch of this corollary is that it works with quantities that are usually not known. For example, the operator $\mathcal{T}$, used to define κ, explicitly depends on the matrix $W+E$. However, often not the perturbation E itself but only an upper bound on its norm is given. For this reason, given a partitioning (2.3), let us use the unperturbed data to define an operator $\mathcal{T}_W : \operatorname{dom}\mathcal{T} \to \operatorname{codom}\mathcal{T}$ as follows:

$$\mathcal{T}_W : (P,Q) \mapsto \begin{bmatrix} P \\ Q \end{bmatrix} A_{11} - \begin{bmatrix} P \\ Q \end{bmatrix} \begin{bmatrix} A_{22} & G_{22} & -G_{12}^T \\ H_{22} & A_{22}^T & A_{12}^T \\ 0 & 0 & A_{11}^T \end{bmatrix}. \tag{5.2}$$

The operator $\mathcal{T}_W^\dagger$ and its norm are defined in the same sense as $\mathcal{T}^\dagger$ and $\|\mathcal{T}^\dagger\|$.

THEOREM 5.2. *Let $U=[X,Z,J^TX,J^TZ]$ be orthogonal symplectic, and suppose that $\mathcal{X}=\operatorname{span} X$ is a semisimple isotropic invariant subspace of the skew-Hamiltonian matrix W so that*

$$U^TWU = \begin{bmatrix} A_{11} & A_{12} & G_{11} & G_{12} \\ 0 & A_{22} & -G_{12}^T & G_{22} \\ 0 & 0 & A_{11}^T & 0 \\ 0 & H_{22} & A_{12}^T & A_{22}^T \end{bmatrix}. \tag{5.3}$$

Given a skew-Hamiltonian perturbation E, let

$$U^TEU = \begin{bmatrix} E_{11} & E_{12} & E_{13} & E_{14} \\ E_{21} & E_{22} & -E_{14}^T & E_{24} \\ E_{31} & E_{32} & E_{11}^T & E_{21}^T \\ -E_{32}^T & E_{42} & E_{12}^T & E_{22}^T \end{bmatrix}.$$

Assume that $\hat{\delta}=\sqrt{3}\|\mathcal{T}_W^\dagger\|\cdot\|E\|_F<1$, where $\mathcal{T}_W^\dagger$ is defined by (5.2). *Set*

$$\hat{\gamma} = \left\| \begin{bmatrix} E_{21} \\ E_{31} \\ E_{32}^T \end{bmatrix} \right\|_F, \quad \hat{\eta} = \left\| \begin{bmatrix} A_{12} & G_{11} & G_{12} \\ A_{22} & -G_{12}^T & G_{22} \\ H_{22} & A_{12}^T & A_{22}^T \end{bmatrix} \right\|_F + \left\| \begin{bmatrix} E_{12} & E_{13} & E_{14} \\ E_{22} & -E_{14}^T & E_{24} \\ E_{42} & E_{12}^T & E_{22}^T \end{bmatrix} \right\|_F,$$

and $\hat{\kappa}=\|\mathcal{T}_W^\dagger\|/(1-\hat{\delta})$. Then if

$$8\hat{\gamma}\hat{\kappa}<1, \qquad 20\hat{\gamma}\hat{\eta}\hat{\kappa}^2<1,$$

there are matrices P and Q satisfying

$$\|(P,Q)\|_F \le 2\hat{\gamma}\hat{\kappa}+4\hat{\gamma}^2\hat{\kappa}^2 < 2.5\hat{\gamma}\hat{\kappa}$$

so that the columns of

$$\hat{X} = \left(X + [Z,J^TZ,J^TX] \begin{bmatrix} P \\ Q \end{bmatrix} \right) (I+P^TP+Q^TQ)^{-1/2}$$

form an orthonormal basis for an isotropic invariant subspace of $\hat{W}=W+E$.

Proof. First, note that the semisimplicity of $\mathcal{X}$ implies that $\mathcal{T}_W$ is onto. The operator $\tilde{\mathcal{T}}$, defined in section 4.2, is decomposed into $\mathcal{T}_W - \triangle\mathcal{T}_W$, where $\triangle\mathcal{T}_W$: dom $\mathcal{T} \to$ codom $\mathcal{T}$ is given by

$$\triangle\mathcal{T}_W : \begin{bmatrix} P \\ R \end{bmatrix} \mapsto \begin{bmatrix} P \\ R \end{bmatrix} E_{11} - \begin{bmatrix} E_{22} & E_{24} & -E_{14}^T \\ E_{42} & E_{22}^T & E_{12}^T \\ -E_{32} & -E_{21}^T & E_{11}^T \end{bmatrix} \begin{bmatrix} P \\ R \end{bmatrix} - \begin{bmatrix} 0 \\ F \end{bmatrix}$$

with $F = P^T[\begin{smallmatrix} E_{32}^T \\ E_{21} \end{smallmatrix}]$. Hence, $\|\triangle\mathcal{T}_W\| \le \sqrt{3}\|E\|_F$, and Lemma 4.4 implies that

$$\tilde{\mathcal{T}}^{\tilde{\dagger}} = \sum_{i=0}^{\infty} (\mathcal{T}_W^{\dagger} \circ \triangle\mathcal{T}_W)^i \circ \mathcal{T}_W^{\dagger}$$

converges absolutely and satisfies $\|\tilde{\mathcal{T}}^{\tilde{\dagger}}\| \le \hat{\kappa}$. The remainder of the proof is analogous to the proof of Theorem 4.6. ∎

The bound (5.1) on the canonical angles between $\mathcal{X}$ and $\hat{\mathcal{X}}$ holds with the quantities γ and κ replaced by $\hat{\gamma}$ and $\hat{\kappa}$:

$$\|\tan\Theta(\mathcal{X}, \hat{\mathcal{X}})\|_F \le 2\hat{\gamma}\hat{\kappa} + 4\hat{\gamma}^2\hat{\kappa}^2 < 2.5\hat{\gamma}\hat{\kappa}. \tag{5.4}$$

Similar to the standard notion of the condition number for an invariant subspace of a general matrix [16], we define the *structured condition number* $c_W(\mathcal{X})$ *for a semisimple isotropic invariant subspace* $\mathcal{X}$ *of a skew-Hamiltonian matrix* by the quantity that is approximated by $\|\Theta(\mathcal{X}, \hat{\mathcal{X}})\|_F/\varepsilon$ as the perturbation level ε tends to zero. From the bound (5.4) and the expansion of $\tan(\cdot)$ around zero we conclude that $c_W(\mathcal{X})$ satisfies

$$c_W(\mathcal{X}) := \lim_{\varepsilon \to 0} \sup_{\substack{\|E\|_F \le \varepsilon \\ E \text{ skew-Hamiltonian}}} \frac{\|\Theta(\mathcal{X}, \hat{\mathcal{X}})\|_F}{\varepsilon} \le \alpha \|\mathcal{T}_W^{\dagger}\|$$

for some $\alpha \le 2$. The presence of the factor α in this bound is artificial; a slight modification of the proof of Theorem 4.6 shows that α can be made arbitrarily close to 1 under the assumption that the perturbation E is sufficiently small. This reveals that $\|\mathcal{T}_W^{\dagger}\|$ is an upper bound on $c_W(\mathcal{X})$.

To show that $c_W(\mathcal{X})$ and $\|\mathcal{T}_W^{\dagger}\|$ actually coincide we construct skew-Hamiltonian perturbations E so that

$$\lim_{\|E\|_F \to 0} \|\Theta(\mathcal{X}, \hat{\mathcal{X}})\|_F / \|E\|_F \ge \|\mathcal{T}_W^{\dagger}\|$$

holds. Given a block Schur decomposition of the form (5.3), choose matrices E_{21} and E_{31} so that $\|(E_{21}, E_{31})\|_F = 1$ and $\|\mathcal{T}_W^{\dagger}(E_{21}, E_{31})\|_F = \|\mathcal{T}_W^{\dagger}\|$, and consider the perturbation

$$E = \varepsilon \cdot [Z, J^T X, J^T Z] \begin{bmatrix} E_{21} \\ E_{31} \end{bmatrix} X^T.$$

By choosing ε sufficiently small, we may assume that there is an invariant subspace $\hat{\mathcal{X}}$ of $W + E$ satisfying $\|\Theta(\mathcal{X}, \hat{\mathcal{X}})\|_2 < \frac{\pi}{2}$. This implies the existence of matrices P and Q so that the columns of

$$\hat{X} = \left(X + [Z, J^T Z, J^T X] \begin{bmatrix} P \\ Q \end{bmatrix} \right) (I + P^T P + Q^T Q)^{-1/2}$$

form an orthonormal basis of $\hat{\mathcal{X}}$. We have seen that any such matrix pair (P, Q) must satisfy the nonlinear matrix equation

$$\mathcal{T}_W(P,R) - \triangle\mathcal{T}_W(P,R) + \Phi(P,R) = \varepsilon \begin{bmatrix} E_{21} \\ E_{31} \end{bmatrix}, \tag{5.5}$$

where R, $\triangle\mathcal{T}_W$, and Φ are defined as in (4.1), (4.9), and (4.14), respectively. If we decompose

$$(P,R) = (P_1 + P_2, R_1 + R_2), \quad (P_1,R_1) \in \operatorname{kernel}(\mathcal{T}_W),\ (P_2,R_2) \in \operatorname{kernel}(\mathcal{T}_W)^{\perp},$$

then

$$(P_2,R_2) = \varepsilon \cdot \mathcal{T}_W^{\dagger}(E_{21},E_{31}) + \mathcal{T}_W^{\dagger} \circ [\triangle\mathcal{T}_W(P,R) - \Phi(P,R)].$$

Since $\|(P,R)\| = \mathcal{O}(\varepsilon)$, it follows that $\|\triangle\mathcal{T}_W(P,R) - \Phi(P,R)\|_F = \mathcal{O}(\varepsilon^2)$ and thus

$$\lim_{\varepsilon\to 0} \|(P_2,R_2)\|_F/\varepsilon = \|\mathcal{T}_W^{\dagger}(E_{21},E_{31})\|_F = \|\mathcal{T}_W^{\dagger}\|.$$

Combining this equality with $\|(P,R)\|_F \geq \|(P_2,R_2)\|_F$ and $\|\Theta(\mathcal{X},\hat{\mathcal{X}})\|_F = \|(P,R)\|_F + \mathcal{O}(\varepsilon^2)$ yields the desired result:

$$\lim_{\varepsilon\to 0} \|\Theta(\mathcal{X},\hat{\mathcal{X}})\|_F/\varepsilon \geq \|\mathcal{T}_W^{\dagger}\|.$$

6. On the computation of $\|\mathcal{T}_W^{\dagger}\|$. The discussion above shows that $\|\mathcal{T}_W^{\dagger}\|$ measures the sensitivity of an isotropic invariant subspace. It remains to compute this quantity. It turns out that $\|\mathcal{T}_W^{\dagger}\|$ is considerably easy to compute if $k = 1$ (real eigenvectors).

LEMMA 6.1. *Let $\lambda \in \mathbb{R}$ be an eigenvalue of the skew-Hamiltonian matrix W with algebraic multiplicity two, and let x be an associated eigenvector satisfying $\|x\|_2 = 1$. Given a partitioning of the form* (5.3) *with respect to x, it follows that*

$$\|\mathcal{T}_W^{\dagger}\| = \sigma_{\min}(W_\lambda)^{-1},$$

where $\sigma_{\min}$ denotes the minimum singular value of a matrix and

$$W_\lambda = \begin{bmatrix} A_{22} - \lambda I & G_{22} & -G_{12}^T \\ H_{22} & A_{22}^T - \lambda I & A_{12}^T \end{bmatrix}.$$

Proof. The operator $\mathcal{T}_W$ can be identified with $\left[\begin{smallmatrix} W_\lambda \\ 0 \end{smallmatrix}\right]$. Hence,

$$\|\mathcal{T}_W^{\dagger}\| = \sup_{\|x\|_2=1} \|\mathcal{T}_W^{\dagger}(x,0)\|_2 = \sup_{\|x\|_2=1} \|W_\lambda^{\dagger} x\|_2 = \sigma_{\min}(W_\lambda)^{-1},$$

using the fact that the space of 1×1 skew-symmetric matrices is $\{0\}$. ∎

If $U^T W U$ is in skew-Hamiltonian Schur form (2.2), then $H_{22} = 0$ and A_{22} is in real Schur form. Then the computation of $\|\mathcal{T}_W^{\dagger}\|$ becomes particularly cheap. Construct an orthogonal matrix Q so that

$$W_\lambda Q = \begin{bmatrix} T_{11} & T_{12} & 0 \\ 0 & T_{22}^T & 0 \end{bmatrix}$$

with upper triangular matrices T_{11} and T_{22}. Since Q can be represented as a product of $\mathcal{O}(n)$ Givens rotations (see [5, sect. 12.5]), the computation of T_{11}, T_{12}, and T_{22} requires $\mathcal{O}(n^2)$ floating point operations (flops). In this case, one of the condition number estimators for triangular matrices [6, Chap. 15] can be used to estimate

$$\left\| \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22}^T \end{bmatrix}^{-1} \right\|_2 = \sigma_{\min}(W_\lambda U)^{-1} = \sigma_{\min}(W_\lambda)^{-1}$$

within $\mathcal{O}(n^2)$ flops.

The case $k > 1$ is more complicated. A possible but quite expensive option is provided by the Kronecker product approach that was already used in the proof of Lemma 4.3. Let

$$K_{\mathcal{T}_W} := A_{11}^T \otimes I - I \otimes \begin{bmatrix} A_{22} & G_{22} & -G_{12}^T \\ H_{22} & A_{22}^T & A_{12}^T \\ 0 & 0 & A_{11}^T \end{bmatrix},$$

and let the columns of K_{Skew} form an orthonormal basis for all vectors in vec(codom $\mathcal{T}$). Then $\|\mathcal{T}_W^\dagger\|$ is given by the minimum singular value of the matrix $K_{\mathrm{Skew}}^T K_{\mathcal{T}_W}$. Note that this is a $(2nk - k(3k+1)/2) \times (2nk - k^2)$ matrix, and thus a direct method for computing its minimum singular value requires $\mathcal{O}(k^3 n^3)$ flops.

Another approach would consist of adapting a condition estimator for Sylvester equations [3, 9] to estimate $\|\mathcal{T}_W^\dagger\|$. This would require the application of $\mathcal{T}_W^\dagger$ (and its dual) to particular elements of codom $\mathcal{T}$ (and dom $\mathcal{T}$). The efficient and reliable computation of these "matrix-vector products" is a delicate task (see, e.g., [7]) and is beyond the scope of this paper.

7. Numerical example. Algorithms for computing the derived condition numbers for eigenvectors of skew-Hamiltonian matrices have been implemented in Fortran 77. They are part of HAPACK [1], a prospective software library for solving eigenvalue problems with Hamiltonian, skew-Hamiltonian, or block cyclic structures. Let us illustrate their use with the following $2n \times 2n$ skew-Hamiltonian matrix:

$$W_n = \left[\begin{array}{cccc|cccc} 0 & -1 & \cdots & -1 & 0 & 1 & \cdots & 1 \\ 0 & -1 & \cdots & -1 & -1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & -1 & -1 & \cdots & -1 & 0 \\ \hline 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & \vdots & -1 & -1 & \ddots & \vdots \\ \vdots & & & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & -1 & -1 & \cdots & -1 \end{array}\right].$$

We computed exact values of $\|\mathcal{T}_W^\dagger\|$ for the eigenvector e_1 of W_n, $n = 2, \ldots, 30$. Furthermore, we applied the algorithm proposed in section 6 to produce estimates of $\|\mathcal{T}_W^\dagger\|$. These theoretical results were compared with practical observations in the following way. A skew-Hamiltonian matrix E with random entries chosen from $N(0,1)$ had been scaled so that $\|E\|_F = 10^{-10}$. Using HAPACK routines, we computed eigenvectors v and w corresponding to two identical eigenvalues of $W_n + E$ that have smallest absolute value. Let the columns of U form an orthonormal basis for

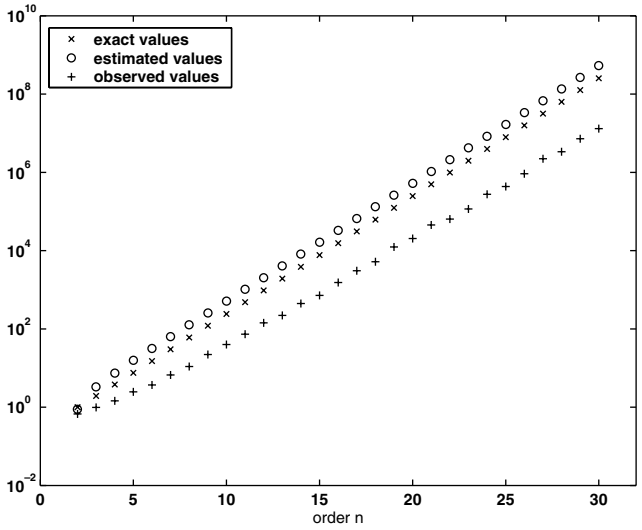


FIG. 7.1. *Exact, estimated and observed values of* $\|\mathcal{T}_W^\dagger\|$ *for the eigenvector* e_1 *of* W_n.

$\text{span}\{v, w\}^\perp$. Then the sine of the angle between $\text{span}\{e_1\}$ and $\text{span}\{v, w\}$ is given by $\|U^H e_1\|_2$. The observed value of $\|\mathcal{T}_W^\dagger\|$ was taken as the maximum over all quantities $10^{10} \cdot \|U^H e_1\|_2$ for 500 different samples of E. The results of the computations, which were performed in a Compaq Visual Fortran environment, are displayed in Figure 7.1. It turns out that the exact value of $\|\mathcal{T}_W^\dagger\|$ is underestimated for $n = 2$ by a factor of 0.88 and overestimated for all other values of n by a factor of at most 2.2. Furthermore, the exact value is consistently larger than the observed value by a factor of at most 20.

8. Conclusions. While the change of eigenvalues under structured perturbations has received a lot of attraction (for a recent work in this area, see, e.g., [17]), invariant subspaces have been much less studied. An extensive perturbation analysis for (block) Hamiltonian Schur forms has been presented in [10]. However, we are not aware of any work on perturbation theory for eigenvectors or invariant subspaces of skew-Hamiltonian matrices. Therefore, we believe that our results are novel. The obtained condition numbers reflect the actual sensitivity of isotropic invariant subspaces rather well, at least for the numerical example presented in the previous section. We hope that the integration of these condition numbers in HAPACK [1] will show whether their usefulness stands the test of practical applications.

Acknowledgments. The author thanks Dr. Michael Karow, Prof. Volker Mehrmann, and Prof. Ji-guang Sun for useful discussions. This work was carried out while the author was at CESAME, Université Catholique de Louvain. The hospitality of this institute is gratefully acknowledged.

REFERENCES

[1] P. BENNER AND D. KRESSNER, *Fortran* 77 *subroutines for computing the eigenvalues of Hamiltonian matrices* II, submitted; see also http://www.math.tu-berlin.de/~kressner/hapack/, 2004.

[2] A. BUNSE-GERSTNER, *Matrix factorizations for symplectic QR-like methods*, Linear Algebra Appl., 83 (1986), pp. 49–77.

[3] R. Byers, *A LINPACK-style condition estimator for the equation* $AX - XB^T = C$, IEEE Trans. Automat. Control, 29 (1984), pp. 926–928.
[4] H. Faßbender, D. S. Mackey, N. Mackey, and H. Xu, *Hamiltonian square roots of skew-Hamiltonian matrices*, Linear Algebra Appl., 287 (1999), pp. 125–159.
[5] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[6] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
[7] A. S. Hodel and P. Misra, *Least-squares approximate solution of overdetermined Sylvester equations*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 279–290.
[8] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
[9] B. Kågström and L. Westin, *Generalized Schur methods with condition estimators for solving the generalized Sylvester equation*, IEEE Trans. Automat. Control, 34 (1989), pp. 745–751.
[10] M. Konstantinov, V. Mehrmann, and P. Petkov, *Perturbation analysis of Hamiltonian Schur and block-Schur forms*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 387–424.
[11] V. Mehrmann and D. Watkins, *Structure-preserving methods for computing eigenpairs of large sparse skew-Hamiltonian/Hamiltonian pencils*, SIAM J. Sci. Comput., 22 (2001), pp. 1905–1925.
[12] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.
[13] J. Stefanovski and K. Trenčevski, *Antisymmetric Riccati matrix equation*, in 1st Congress of the Mathematicians and Computer Scientists of Macedonia (Ohrid, 1996), Sojuz. Mat. Inform. Maked., Skopje, 1998, pp. 83–92.
[14] G. W. Stewart, *Error bounds for approximate invariant subspaces of closed linear operators*, SIAM J. Numer. Anal., 8 (1971), pp. 796–808.
[15] G. W. Stewart, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Rev., 15 (1973), pp. 727–764.
[16] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, New York, 1990.
[17] F. Tisseur, *A chart of backward errors for singly and doubly structured eigenvalue problems*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 877–897.
[18] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.
[19] C. F. Van Loan, *A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix*, Linear Algebra Appl., 61 (1984), pp. 233–251.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 4, pp. 962–984

A TECHNIQUE FOR ACCELERATING THE CONVERGENCE OF RESTARTED GMRES*

A. H. BAKER†, E. R. JESSUP‡, AND T. MANTEUFFEL§

Abstract. We have observed that the residual vectors at the end of each restart cycle of restarted GMRES often alternate direction in a cyclic fashion, thereby slowing convergence. We present a new technique for accelerating the convergence of restarted GMRES by disrupting this alternating pattern. The new algorithm resembles a full conjugate gradient method with polynomial preconditioning, and its implementation requires minimal changes to the standard restarted GMRES algorithm.

Key words. GMRES, iterative methods, Krylov subspace, restart, nonsymmetric linear systems

AMS subject classification. 65F10

DOI. 10.1137/S0895479803422014

1. Introduction.

Iterative methods are a common choice for solving the large sparse system of linear equations

$$Ax = b, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ is nonsingular and $x, b \in \mathbb{R}^n$. A popular class of iterative methods are Krylov subspace methods. Krylov subspace methods find an approximate solution

$$x_i \in x_0 + K_i(A, r_0), \tag{2}$$

where $K_i(A, r_0) \equiv span\{r_0, Ar_0, \ldots, A^{i-1}r_0\}$ denotes an i-dimensional Krylov subspace, x_0 is the initial guess, and r_0 is the initial residual ($r_0 \equiv b - Ax_0$). Krylov subspace methods are also known as polynomial methods since (2) implies that the residual r_i can be written in terms of a polynomial in A: $r_i = p(A)r_0$.

At present, a large variety of Krylov subspace methods exist. When A is nonsymmetric, choosing the most appropriate method can be difficult (see, e.g., [22]), though the generalized minimum residual algorithm (GMRES) [27] is arguably the

*Received by the editors January 24, 2003; accepted for publication (in revised form) by Z. Strakoš July 30, 2004; published electronically May 6, 2005.

http://www.siam.org/journals/simax/26-4/42201.html

†Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Box 808 L-551, Livermore, CA 94551 (abaker@llnl.gov). The work of this author was primarily supported by the Department of Energy Computational Science Graduate Fellowship Program of the Office of Scientific Computing and Office of Defense Programs in the Department of Energy under contract DE-FG02-97ER25308. Portions of this work were performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract W-7405-Eng-48.

‡Department of Computer Science, University of Colorado, Boulder, CO 80309-0430 (jessup@cs.colorado.edu). The work of this author was supported by the National Science Foundation under grant ACI-0072119.

§Department of Applied Mathematics, University of Colorado, Boulder, CO 80309-0526 (tmanteuf@colorado.edu). The work of this author was supported by the National Institute of Health under grant 1-R01-EY12292-01, the National Science Foundation under grant DMS–0084438, the Department of Energy under grant DE-FG03-94ER25217 and DE-FC02-01ER25479, and the National Science Foundation under VIGRE grant DMS-9810751.

most popular choice. GMRES is often referred to as an "optimal" method because it finds the approximate solution in the Krylov subspace that minimizes the 2-norm of the residual [27].

At each iteration of GMRES, the amount of storage and computational work required increases. Therefore, when the required resources make standard GMRES impractical, the restarted version of the algorithm is used as suggested in [27]. In restarted GMRES (GMRES(m)), the method is "restarted" once the Krylov subspace reaches dimension m, and the current approximate solution becomes the new initial guess for the next m iterations. The restart parameter m is generally chosen small relative to n to keep storage and computation requirements reasonable. However, choosing an appropriate restart parameter can be difficult as the choice can significantly affect the convergence rate (see, e.g., [17, 13]).

In general, restarting slows the convergence of GMRES. When an iterative approach is restarted, the current approximation space is discarded at each restart. Therefore, a well-known drawback of GMRES(m) is that orthogonality to previously generated subspaces is not preserved at each restart. In fact, GMRES(m) can stall as a result. Stalling means that there is no decrease in the residual norm at the end of a restart cycle. Restarting also negates the potential for superlinear convergence behavior [29].

This paper is organized as follows. In section 2, we describe some existing modifications to GMRES(m) aimed at accelerating convergence or overcoming stalling. We introduce our new acceleration technique in section 3. We present numerical results and discuss the convergence behavior of the new algorithm in section 4. We close with concluding remarks in section 5.

2. Background. In this section, we briefly describe some existing modifications to the standard GMRES algorithm. These modifications all have the common goal of enhancing the robustness of restarted GMRES. Two primary categories of modification include hybrid iterative methods and acceleration techniques. Hybrid iterative methods combine standard iterative methods in a variety of ways to reduce the number of required vector operations. Many of these methods are essentially modifications to GMRES(m) aimed at improving its performance. Nachtigal, Reichel, and Trefethen provide a general overview of this class of iterative methods in [21]. Our work falls into the category of acceleration techniques. These techniques attempt to mimic the convergence of full GMRES more closely or to accelerate the convergence of GMRES(m) by retaining some of the information that is typically discarded at the time of restart. In [11], Eiermann, Ernst, and Schneider present a thorough overview and analysis of the most common acceleration techniques.

Augmented methods are a class of acceleration techniques. In particular, these methods seek to avoid stalling by improving information in GMRES at the time of the restart. Typically a (nearly) A-invariant subspace is appended to the Krylov approximation space, resulting in an "augmented Krylov subspace" [5]. The invariant subspace of A associated with the smallest eigenvalues is commonly used, as those eigenvalues are thought to hinder convergence the most. Algorithms that include spectral information at the restart to overcome stalling are presented by Morgan in [18], [19], and [20] (GMRES-E, GMRES-IR, and GMRES-DR, respectively) and are further discussed in [5] and [26]. These augmentation techniques are more suitable for some types of problems than others. They can be very effective when convergence is being hampered by a few eigenvalues [18]. However, they may have little effect on highly nonnormal problems [5], or solving the eigenvalue problem may be too costly for the

technique to be beneficial [18]. Of interest to us is the simple framework provided for appending (non-Krylov) vectors to the approximation space.

Another class of acceleration techniques is based on the fact that ideally the approximation space should contain the correction c such that $x = x_0 + c$ is the exact solution to the problem [11]. The nested Krylov subspace method GMRESR (GMRES Recursive) [30] is one such technique. In GMRESR, the outer generalized conjugate residual method (GCR) [12] invokes an inner iterative method (like GMRES) at each step i to approximate the solution to $Ac = r_i$, where r_i is the current residual at step i. The approximate solution to $Ac = r_i$ then becomes the next direction for the outer approximation space. The goal of this method is to obtain similar convergence to that of full GMRES with less computational cost under certain conditions. Note that the method FGMRES (Flexible GMRES) [24] can also be viewed as a method that approximates solutions to similar residual equations at each step. In fact, both FGMRES and GCR provide a framework for using a GMRES-like method with *any* approximation space.

Another related acceleration technique is GCRO (GCR with inner orthogonalization) [7]. The aim of this method is twofold: to compensate for the information that is lost due to restarting as well as to overcome some of the stalling problems that GMRESR can experience in the inner iteration. GCRO is a modification to GMRESR such that the inner iterative method maintains $A^T A$-orthogonality to the outer approximation space. Thus, the approximation from the inner iteration at step i takes into account both the inner and outer approximation spaces. See also [9] for more details on preserving orthogonality in the inner iteration of a nested Krylov subspace method. In most cases, both GCRO and GMRESR must be truncated to keep storage costs reasonable. Therefore, a truncated version of GCRO, the method GCROT (GCRO Truncated), is subsequently described in [8]. GCROT attempts to determine which subspace of the outer approximation space should be retained for the best convergence of future iterations as well as if any portion of the inner Krylov subspace should also be kept.

As Fokkema, Sleijpen, and van der Vorst point out in [15], "the distinction between preconditioning and acceleration is not a clear one." These acceleration techniques (GMRESR, GCRO, and FGMRES) can also be viewed as methods with variable preconditioning (allowing the preconditioner to change with each iteration step). We show that our new method can also be viewed in this way.

3. A new algorithm: LGMRES. In this section, we describe a new method for accelerating GMRES(m). We begin with observations about the convergence behavior of GMRES(m) that lead us to the new technique. We then present the new algorithm LGMRES (Loose GMRES), discuss some of its properties, and compare LGMRES to closely related existing acceleration techniques.

3.1. Motivation. Consider restarted GMRES(m) when solving problem (1). In this discussion, we refer to the group of m iterations between successive restarts as a cycle. The restart number is denoted with a subscript: r_i is the residual after i cycles or $m \times i$ iterations. The residual at the end of cycle $i+1$ is a polynomial in A times the residual from the previous cycle, $r_{i+1} = p_{i+1}^m(A) r_i$, where $p_{i+1}^m(A)$ is the degree m residual polynomial. During each restart cycle (i), GMRES(m) finds $x_{i+1} \in x_i + K_m(A, r_i)$ such that $r_{i+1} \perp AK_m(A, r_i)$ (see, e.g., [25]).

As previously mentioned, GMRES(m) does not maintain orthogonality between approximation spaces generated at successive restarts. As a result, slow convergence or even stalling can occur. In the case of slow convergence, we have observed a pattern

TABLE 1
Results for GMRES(30). Problem size, iterations required for $\|r_i\|_2/\|r_0\|_2 \leq 10^{-9}$, median skip angle, and median sequential angle are listed for each problem.

Problem	Size (n)	Iterations	Median seq. angle $\angle(r_i, r_{i-1})$	Median skip angle $\angle(r_{i+1}, r_{i-1})$
add20	2395	1002	51.3	5.4
orsirr_1	1030	6659	23.0	6.9
orsreg_1	2205	888	59.3	8.4
sherman_1	1000	3688	27.5	.2

in GMRES(m) where the residual vectors point in nearly the same direction at the end of every other restart cycle. In other words, the angle between r_{i+1} and r_{i-1} is small and $r_{i+1} \approx \alpha\, r_{i-1}$. We refer to the angles between *every other* residual vector as *skip* angles, e.g., $\angle(r_{i+1}, r_{i-1})$, and the angles between consecutive restart cycles as *sequential* angles.

For many problems, we find that skip angles are relatively small even when the sequential angles are a reasonable size (i.e., stalling is not occurring). For example, Table 1 gives results for GMRES(30) on several problems available from the Matrix Market Collection [23]. The number of iterations required for convergence ($\|r_i\|_2/\|r_0\|_2 \leq 10^{-9}$) as well as the median sequential and median skip angle values are listed. GMRES(30) is not stalling for these four problems. However, the low skip angle values appear to indicate that faster convergence should be possible if some degree of orthogonality to previous approximation spaces were maintained, a goal embraced by several acceleration techniques described in section 2. In our experience, this type of alternating pattern is most pronounced (most "exact") for symmetric matrices, but it is noticeable for many nonsymmetric matrices as well.

There no mechanism in GMRES(m) to prevent this alternating phenomenon because it is simply a symptom of the lack of orthogonality between the approximation space generated during a particular restart cycle of GMRES(m) and the approximation spaces from previous cycles. However, only for the special case when the restart parameter is one less than the matrix order can we show that alternating must occur for both symmetric and skew-symmetric problems. Consider the following lemma.

LEMMA 1 (equivalent constraints). *When $A \in \mathbb{R}^{n\times n}$ is symmetric or skew-symmetric, and w and y are arbitrary real vectors of length n, the requirement that $w \perp AK_m(A, y)$ is equivalent to the requirement that $w \perp A^T K_m(A^T, y)$.*

With this easily proved lemma, the following theorem is straightforward.

THEOREM 2 (alternating residuals). *When $A \in \mathbb{R}^{n\times n}$ is symmetric or skew-symmetric and the restart parameter is one less than the matrix order ($m = n - 1$), GMRES(m) produces a sequence of residual vectors at the end of each restart cycle such that $r_{i+2} = \alpha r_i$, $|\alpha| \leq 1$.*

Proof. During restart cycle i,

$$r_i \perp AK_m(A, r_{i-1}) \Rightarrow r_{i-1} \perp A^T K_m(A^T, r_i).$$

From Lemma 1,

$$r_{i-1} \perp A^T K_m(A^T, r_i) \Rightarrow r_{i-1} \perp AK_m(A, r_i). \tag{3}$$

Let $W_m \equiv [w_1\ w_2\ \ldots\ w_m]$ be an orthonormal basis for $AK_m(A, r_i)$. There exists a w_n such that $W_n = [W_m\ w_n]$ is an orthonormal basis for $\mathbb{R}^n$. From (3), $r_{i-1} = \alpha w_n$,

where α is some scalar. During restart cycle $i+1$,

$$r_{i+1} \perp AK_m(A, r_i) \Rightarrow r_{i+1} = \beta w_n,$$

where β is some scalar. Therefore, $r_{i+1} = \frac{\beta}{\alpha} r_{i-1}$, and $|\frac{\beta}{\alpha}| \leq 1$ because the GMRES(m) residual norm is nonincreasing. □

3.2. Idea and implementation. The motivation for the new algorithm, LGMRES, came from a desire to prevent the alternating behavior observed for GMRES(m) which results in repetitive information in successive restart cycles. In addition, we wanted a method for which the idea and implementation easily lent themselves to a block method for solving a single right-hand side system (see, e.g., [2]). Therefore, the new algorithm is a combination of ideas from several existing acceleration techniques described in section 2: GMRES-E, GMRESR, and GCRO. In short, LGMRES utilizes the simple framework of Morgan's GMRES-E method [18] for appending vectors to the standard Krylov space in a manner that allows for the extension to a block method as in [5], for example. GMRESR [30] and GCRO [7], on the other hand, provide ideas for choosing appropriate vectors to append to the standard Krylov approximation space. The algorithmic components from these existing techniques are combined in a manner that results in a new acceleration technique with both a simple implementation and the ability to prevent the previously described alternating behavior.

To prevent alternating, LGMRES mimics GMRESR's technique of including approximations to the error in the current approximation space. Suppose that $\hat{x}$ is the true solution to problem (1). The error after the ith restart cycle of GMRES(m) is denoted by e_i, where

$$e_i \equiv \hat{x} - x_i. \tag{4}$$

As explicitly pointed out in [11] and noted in section 2, if our approximation space contains the exact correction e_i such that $\hat{x} = x_i + e_i$, then we have solved the problem. We define

$$z_i \equiv x_i - x_{i-1} \tag{5}$$

as the approximation to the error after the ith GMRES(m) restart cycle, and $z_j \equiv 0$ for $j < 1$. This error approximation vector serves as our choice of vector with which to augment our next approximation space $K_m(A, r_i)$. Note that $z_i \in K_m(A, r_{i-1})$. Therefore, this error approximation z_i in some sense represents the space $K_m(A, r_{i-1})$ generated in the previous cycle and subsequently discarded and is a natural choice of vector with which to augment our next approximation space $K_m(A, r_i)$.

We denote our new restarted augmented GMRES algorithm by LGMRES(m, k). LGMRES(m, k) augments the standard Krylov approximation space with k previous approximations to the error. Therefore, at the end of restart cycle $i+1$, LGMRES(m, k) finds an approximate solution to (1) in the following way:

$$x_{i+1} = x_i + q_{i+1}^{m-1}(A) r_i + \sum_{j=i-k+1}^{i} \alpha_{ij} z_j, \tag{6}$$

where polynomial q_{i+1}^{m-1} and α_{ij} are chosen such that $\|r_{i+1}\|_2$ is minimized. Note that $k = 0$ corresponds to standard GMRES(m).

The implementation of LGMRES(m, k) is quite similar to that of Morgan's GMRES with eigenvectors (GMRES-E) method described in [18] and requires minimal changes to the standard GMRES(m) implementation. At each restart cycle (i) we

1. $r_i = b - Ax_i$, $\beta = \|r_i\|_2$, $v_1 = r_i/\beta$, $s = m + k$
2. for $j = 1 : s$
3. $\quad u = \begin{cases} Av_j & \text{if } j \le m, \\ Az_{i-(j-m-1)} & \text{otherwise} \end{cases}$
4. $\quad$ for $l = 1 : j$
5. $\quad\quad h_{l,j} = \langle u, v_l \rangle$
6. $\quad\quad u = u - h_{l,j} v_l$
7. $\quad$ end
8. $\quad h_{j+1,j} = \|u\|_2$, $v_{j+1} = u/h_{j+1,j}$
9. end
10. $V_{s+1} = [v_1, \ldots, v_m, \ldots, v_{m+k+1}]$, $W_s = [v_1, \ldots, v_m, z_i, \ldots, z_{i-k+1}]$, $H_s = \{h_{l,j}\}_{1 \le l \le j+1; 1 \le j \le s}$
11. find y_s s.t. $\|\beta e_1 - H_s y_s\|_2$ is minimized
12. $z_{i+1} = W_s y_s$ (also $Az_{i+1} = V_{s+1} H_s y_s$)
13. $x_{i+1} = x_i + z_{i+1}$

FIG. 1. *LGMRES(m, k) for restart cycle i.*

generate the Krylov subspace $K_m(A, r_i)$ and augment it with the k most recent error approximations z_j, $j = (i - k + 1) : i$. The augmented approximation space $\mathcal{M} = K_m(A, r_i) \cup span\{z_j\}_{j=(i-k+1):i}$ has dimension $s \equiv m + k$. We then find the approximate solution from $\mathcal{M}$ whose corresponding residual is a minimum in the Euclidean norm.

One restart cycle (i) of the LGMRES(m, k) algorithm is given in Figure 1. Note that V_{s+1} is the $n \times (s + 1)$ orthonormal matrix whose first $m + 1$ columns are the Arnoldi vectors and last s columns result from orthogonalizing the k error approximation vectors (z_j, $j = (i - k + 1) : i$) against the previous columns of Arnoldi vectors. W_s is the $n \times s$ matrix whose first m columns are equal to the first m columns of V_{s+1} and whose last k columns of W are the k error approximation vectors (typically normalized so that all columns are of unit length). Then the relationship

$$AW_s = V_{s+1} H_s \tag{7}$$

holds for LGMRES(m, k), where H_s denotes an $(s+1) \times s$ Hessenberg matrix whose elements $h_{l,j}$ are defined in the algorithm in Figure 1. This relationship is analogous to equations (11) in [18] and (3) in [27].

When implementing LGMRES(m, k), only m matrix-vector multiplies are required per restart cycle, irrespective of the value of k, provided that we form both z_i and Az_i at the end of cycle i as is done in the algorithm given in Figure 1. Note that forming Az_i does not require an explicit multiplication by A and that at most k pairs of z_j and Az_j need to be stored. Typically the number of vectors appended, k, is much smaller than the restart parameter m (discussed in section 4). The algorithm requires storage for the following vectors of length n: $m + k + 1$ orthogonal basis vectors ($v_1, v_2, \ldots v_{m+k+1}$), k pairs of z_j and Az_j, the approximate solution, and the right-hand side. Therefore, this implementation of LGMRES(m, k) requires storage for $m + 3k + 3$ vectors of length n and m matrix-vector multiplies per restart cycle. Recall that standard GMRES($m + k$) requires storage for $m + k + 3$ vectors of length n and $m + k$ matrix-vector multiplies per restart cycle (see, e.g., [25]). Also, LGMRES(m, k) and GMRES($m + k$) require equivalent numbers of inner products and vector updates. One could reduce the storage requirement for LGMRES(m, k) by

recomputing Az_i in each cycle. The storage requirement for vectors of length n would then drop to $m+2k+3$, but the number of matrix-vector multiplies required per cycle would increase to $m+k$. We prefer the former method (as given in Figure 1) because it reduces the number of matrix-vector multiplies and is therefore generally faster.

Note that only i error approximations are available at the beginning of restart cycles with $i < k$ because $z_j = 0$ when $j < 1$. Therefore, we recommend using additional Arnoldi vectors instead of z_j when $j < 1$ so that the approximation space is of dimension $m + k$ for each cycle. In other words, the first cycle ($i = 0$) of LGMRES(m, k) is equivalent to the first cycle of GMRES($m + k$).

LGMRES(m, k) can be preconditioned in a straightforward manner. Let M^{-1} denote the preconditioner. For left preconditioning, we simply precondition the initial residual in line 1 of the algorithm in Figure 1 ($r_i = M^{-1}b - M^{-1}Ax_i$). Then we replace A with $M^{-1}A$ everywhere in lines 3 and 12. For right preconditioning, the required modifications are more subtle. To include previous approximations to the error in the approximation space, we must now append $\hat{z}_j \equiv M(x_j - x_{j-1}) = Mz_j$ instead of z_j to the standard Krylov subspace (no matrix-vector products with M are explicitly computed). Therefore, we replace A with AM^{-1} everywhere in lines 3 and 12 and z with $\hat{z}$ everywhere in lines 3, 10, and 12. While no explicit change is required for line 13 as given in Figure 1, note that, with right preconditioning, line 13 is equivalent to $x_{i+1} = x_i + M^{-1}\hat{z}_{i+1}$.

3.3. Properties. In this section, we first address the similarity between LGMRES and a full conjugate gradient (FCG) method with polynomial preconditioning. We then discuss skip angles and sequential angles for both GMRES(m) and LGMRES(m, k).

We consider the "full" (i.e., nontruncated) version of LGMRES, denoted by LGMRES(m), in which *all* previous error approximations are kept (i.e., $k = i$):

$$z_{i+1} = q_{i+1}^{m-1}(A)r_i + \sum_{j=1}^{i} \alpha_{ij} z_j. \tag{8}$$

In this form, the resemblance of LGMRES(m) to a minimal residual FCG method that minimizes $\|e_i\|_{A^TA}$ at each step, such as ORTHOMIN, is readily apparent (see, e.g., [25] or [1]). In (8), the GMRES(m) iteration polynomial ($q_{i+1}^{m-1}(A)$) corresponds to a polynomial preconditioner. Notice, however, that LGMRES effectively changes the preconditioner with each iteration i, whereas preconditioned FCG typically uses a constant preconditioner (not dependent on i). Vectors z_j in (8) correspond to conjugate gradient direction vectors in that they are also A^TA-orthogonal, as is shown below. Therefore, we can categorize the LGMRES(m, k) method as a truncated polynomial-preconditioned FCG method.

THEOREM 3 (orthogonality of the error approximations). *The error approximation vectors $z_j \equiv x_j - x_{j-1}$ with which we augment the Krylov space in full LGMRES* (8) *or truncated LGMRES* (6) *are A^TA-orthogonal.*

Proof. First, we define subspaces $\mathcal{M}_{i+1}$ and $\mathcal{M}_i$ as

$$\mathcal{M}_{i+1} \equiv K_m(A, r_i) \cup span\{z_j\}_{j=(i-k+1):i}$$

and

$$\mathcal{M}_i \equiv K_m(A, r_{i-1}) \cup span\{z_j\}_{j=(i-k):(i-1)},$$

respectively. By construction,

$$r_i \perp A\mathcal{M}_i \qquad \text{and} \qquad r_{i+1} \perp A\mathcal{M}_{i+1}.$$

From (5),

$$r_i - r_{i+1} = Az_{i+1}.$$

Therefore,

$$Az_{i+1} \perp A(\mathcal{M}_i \cap \mathcal{M}_{i+1}).$$

Because $\{z_j\}_{j=(i-k+1):i} \subset \mathcal{M}_i \cap \mathcal{M}_{i+1}$,

$$z_{i+1} \perp_{A^TA} \{z_j\}_{j=(i-k+1):i}. \qquad \square$$

Although full LGMRES is interesting from a theoretical point of view, it is not a practical algorithm. Storing all past values of z_j ($j = 1 : i$) requires an increasing amount of storage at each restart. As with GMRESR and GCRO, truncating is necessary. Therefore, in practice, we use truncated LGMRES(m, k) as given in (6) with some $k < i$. In section 4, we show that optimal values for k are typically very small: $k \leq 3$. Furthermore, note that the A^TA-orthogonality of the error approximation vectors shown in Theorem 3 is not exploited in the implementation of LGMRES described in the previous section. In fact, a total of k vector products and updates per restart cycle in the algorithm given in Figure 1 are extraneous due to a zero vector product in line 5. However, for small k, the benefit of modifying the LGMRES(m, k) implementation to exploit this orthogonality is negligible.

Now we compare the skip and sequential angles for GMRES(m) and LGMRES(m, k). For standard restarted GMRES, the angle between two residuals from consecutive restart cycles (i.e., the sequential angle) can be expressed in terms of a ratio of their residual norms. The following result is mathematically equivalent to a result first given by Simoncini as Proposition 4.1 in [28], but here we present it in a simplified form with a more straightforward and concise proof.

THEOREM 4 (GMRES(m) sequential angles). *Let r_{i+1} and r_i be the residuals from GMRES restart cycles $i + 1$ and i, respectively. Then the angle between these residuals is given by*

$$\cos\angle(r_{i+1}, r_i) = \frac{\|r_{i+1}\|_2}{\|r_i\|_2}. \tag{9}$$

Proof. In restart cycle $i + 1$ of GMRES(m), $x_{i+1} = x_i + \delta_{i+1}$, where $\delta_{i+1} \in K_m(A, r_i)$. Therefore, the corresponding residual is

$$r_{i+1} = r_i - A\delta_{i+1}.$$

By construction,

$$\langle r_{i+1},\, A\delta_{i+1}\rangle = 0 \quad \Rightarrow \quad \langle r_{i+1},\, r_i\rangle = \langle r_{i+1},\, r_{i+1}\rangle = \|r_{i+1}\|_2^2. \tag{10}$$

The above, combined with the definition of cosine, completes the proof. $\square$

The above indicates that, for GMRES(m), the convergence rate correlates to the size of the angles between consecutive residual vectors. If consecutive residual vectors are nearly orthogonal to each other, then convergence is fast. (If we find an r_{i+1} such that $r_{i+1} \perp r_i$, then we have found the exact solution.) Note that this result also holds for LGMRES. We refer to the related work in [10] for a more general discussion on how the angles between approximation and residual spaces define convergence for

Krylov methods. Note that Theorem 4 above is also a special case of the more general result in (3.9) in [10]. Now we consider the angle between every other residual (i.e., the skip angle).

THEOREM 5 (GMRES(m) skip angles). *Let r_{i+1} and r_{i-1} be the residuals from GMRES restart cycles $i+1$ and $i-1$, respectively. Then the angle between these residuals is given by*

$$\cos\angle(r_{i+1}, r_{i-1}) = \frac{\|r_{i+1}\|_2}{\|r_{i-1}\|_2} - \frac{\langle A\delta_{i+1},\ A\delta_i\rangle}{\|r_{i+1}\|_2\|r_{i-1}\|_2},$$

where $r_{i+1} = r_i - A\delta_{i+1}$ and $r_i = r_{i-1} - A\delta_i$.

Proof. As in the previous proof, it is easily shown that

$$\langle r_{i+1},\ r_{i-1}\rangle = \langle r_{i+1},\ r_{i+1}\rangle - \langle A\delta_{i+1},\ A\delta_i\rangle. \tag{11}$$

The proof follows directly from (11). ∎

In terms of describing convergence, the above result is not immediately helpful. However, we will discuss a few of its implications after giving a corresponding result for LGMRES. Recall from section 3.2 that LGMRES(m, k) appends k previous approximations to the error to the current Krylov approximation space. Therefore, if $k \geq 1$, then $r_{i+1} \perp AK_m(A, r_i)$ and $r_{i+1} \perp Az_i$ at the end of restart cycle $i+1$. Since $Az_i = r_{i-1} - r_i$,

$$\langle r_{i+1},\ r_{i-1} - r_i\rangle = 0 \tag{12}$$

after $i+1$ LGMRES cycles, and we can prove the following theorem.

THEOREM 6 (LGMRES: Every other residual vector). *Let r_{i+1} and r_{i-1} be the residuals from LGMRES restart cycles $i+1$ and $i-1$, respectively. Then the angle between these residuals is given by*

$$\cos\angle(r_{i+1}, r_{i-1}) = \frac{\|r_{i+1}\|_2}{\|r_{i-1}\|_2}.$$

Proof. This theorem directly follows from Theorems 5 and 3 (noting the correlation between δ_i in GMRES(m) and z_i in LGMRES). Alternatively, from (12) and (10),

$$\langle r_{i+1},\ r_{i-1}\rangle = \langle r_{i+1},\ r_i\rangle = \langle r_{i+1},\ r_{i+1}\rangle.$$

The proof follows directly from the above relation. ∎

This result indicates that, for LGMRES, the progress of the iteration also correlates with the skip angles. Therefore, fast convergence implies large skip angles. More generally, for any $0 \leq j \leq k$ and $i \geq k$, we can show for LGMRES(m, k) that

$$\cos\angle(r_{i+1}, r_{i-j}) = \frac{\|r_{i+1}\|_2}{\|r_{i-j}\|_2}.$$

When a problem exhibits signs of alternating residuals with GMRES(m), then the angle between r_{i-1} and r_{i+1} is small. In this case, since $A\delta_{i+1} = r_i - r_{i+1}$ and $A\delta_i = r_{i-1} - r_i$, then the term $\langle A\delta_{i+1},\ A\delta_i\rangle$ in Theorem 5 is negative. We have observed this result in our experiments, and it can be seen pictorially in Figure 2. Since LGMRES appends a previous error approximation to the approximation space

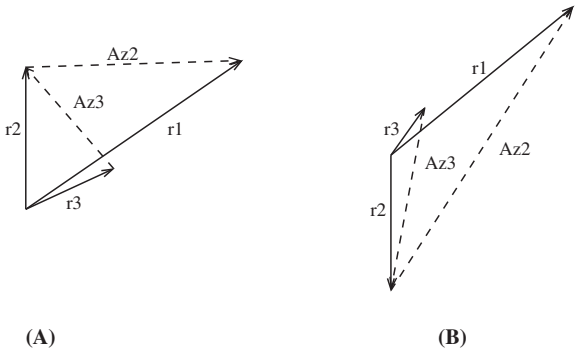


FIG. 2. *Two cases with alternating residual vectors:* r_1 *and* r_3 *point in nearly the same direction.* $\langle Az_3,\ Az_2\rangle < 0$ *in both* (A) *and* (B).

during cycle $i+1$, the term $\langle A\delta_i,\ A\delta_{i-1}\rangle$ is equal to zero by construction. We show in section 4.1 that this LGMRES augmenting scheme tends to increase the skip angle over that of GMRES(m) and prevents the alternating behavior often observed in restarted GMRES.

We also investigated adaptive versions of LGMRES that determine whether or not to augment during each restart cycle. One often effective adaptive version is based on the above observation that the term $\langle A\delta_{i+1},\ A\delta_i\rangle$ in Theorem 5 is generally negative when alternating occurs. In particular, after m standard Arnoldi iterations in restart cycle $i+1$, we form the current residual $\hat{r}_{i+1}$. In the $k=1$ case, the decision is made to augment during cycle $i+1$ if $\langle \hat{r}_{i+1},\ A\delta_i\rangle > 0$. Referring back to Theorem 5, note that $\langle r_{i+1},\ A\delta_i\rangle = -\langle A\delta_{i+1},\ A\delta_i\rangle$. Results for this adaptive version of LGMRES are discussed in section 4.1.

3.4. Comparison to existing methods. As previously stated, LGMRES(m, k) acts as an accelerator for GMRES(m). The algorithm is not designed to overcome stalling as the error approximation vectors, z_j, are zero when the residual norm does not decrease within a cycle. Thus, while the LGMRES implementation mimics that of Morgan's GMRES-E [18], we do not compare the two algorithms, as GMRES-E is most effective for problems that stall due to the effects of a few eigenvalues. However, as noted at the beginning of this section, the general idea of LGMRES is very similar to that of GCRO [7]; both methods look for a minimum residual solution in the approximation space consisting of previous approximations to the error as well as a Krylov space built on the current residual. The algorithms are not mathematically equivalent, and we briefly explain their similarities and differences in this section. First, we discuss the GMRESR [30] method, of which GCRO is a modification. Then the theoretical differences between (nontruncated) GCRO and full LGMRES are briefly described, followed by a comparison of the two truncated algorithms GCROT and LGMRES(m, k).

The nested Krylov subspace methods GMRESR and GCRO consist of an outer GCR method that invokes an inner GMRES method at each iteration to find an approximation to the error. Generally a fixed number of GMRES steps are taken at each inner iteration, say, m. GCR is a minimum residual method that maintains two bases, U_i and $C_i = AU_i$, where $C_i^T C_i = I_i$. Typically U_i is an $A^T A$-orthogonal basis for the Krylov space $K_i(A, r_0)$. However, the implementation of GCR is such that U_i can actually contain any vectors (i.e., range$(U_i) \neq K_i(A, r_0)$) [7]. In particular,

in both the GMRESR and GCRO methods, range(U_i) contains all of the previous approximations to the error from the inner GMRES method.

GMRESR is essentially performing two separate minimizations: one over the inner GMRES approximation space to find a new error approximation and one over the outer approximation space (consisting of the new error approximation and all previous error approximations) to update the current global approximate solution. The clever improvement of GCRO over GMRESR is that the GCRO minimization in the inner iteration takes into account the outer approximation space. In other words, the two methods are not mathematically equivalent, and GCRO solves the following minimization problem at each inner iteration:

$$\min\|b - Ax_{i+1}\|_2 \quad \text{s.t.} \quad x_{i+1} \in \text{range}(U_i) \oplus \text{range}(W_m), \tag{13}$$

where W_m is an orthogonal basis for $K_m(A_C, r_i)$ generated by the inner GMRES method and $A_C \equiv (I - C_i C_i^T)A$. The Krylov space $K_m(A_C, r_i)$ is a result of GCRO maintaining orthogonality against C_i from the beginning of the Arnoldi iteration, and W_{m+1} satisfies $W_{m+1} \perp \text{range}(C_i)$. Thus, when r_i is projected onto AW_m resulting in new residual r_{i+1}, that new residual is also orthogonal to range(C_i) as desired. The solution to the global minimization problem of (13) is then found.

Similarly to GCRO, full LGMRES finds a minimum residual solution in an approximation space consisting of all previous error approximations (z_j) together with a Krylov space built off the current residual:

$$\min\|b - Ax_{i+1}\|_2 \quad \text{s.t.} \quad x_{i+1} \in \text{range}(Z_i) \oplus \text{range}(V_m),$$

where V_m is an orthogonal basis for $K_m(A, r_i)$ and $Z_i \equiv [z_1 \dots z_i]$. In the case of LGMRES, the Arnoldi iteration does not maintain orthogonality against the previous error approximations. Instead, the error approximations are simply appended onto the generated Krylov subspace, which leads to a greater number of orthogonalizations than for GCRO if k is large.

The difference in generation of the Krylov subspaces is a subtle difference between GCRO and LGMRES. Matrices A_C and A do not generate equivalent residual spaces ($A_C K_m(A_C, r_i)$ and $AK_m(A, r_i)$, respectively). See [16] for more on matrices that generate equivalent Krylov residual spaces. Therefore, the residual projected onto these spaces is not the same unless the unlikely situation occurs where range(V_m) $\perp$ range(C_i). Finally we remark that as with GCRO, LGMRES is also not equivalent to GMRESR since the error approximation vectors are determined by a single minimization over the global space consisting of previous approximations to the error as well as a Krylov space built on the current residual.

GCROT [8] is a more practical truncated version of GCRO. GCROT truncates the outer approximation space by examining angles between subspaces and determining which subspaces (not vectors) are important for convergence. It is assumed that if a subspace was important for past convergence, then it will be important for future convergence and should be retained. Similarly, vectors from the inner GMRES iteration may also be kept. The implementation of GCROT(m, k_{max}, k_{new}, s, p_1, p_2) requires specification of six different parameters that affect the truncation.

LGMRES(m, k), on the other hand, is truncated in a more obvious manner, retaining only the most recent k error approximation vectors. For ORTHOMIN, it has been observed that truncating the recursion such that only one or two previous direction vectors are retained is quite effective when A is nearly symmetric [31]. Therefore, we attribute the effectiveness of the LGMRES method's naive truncation

strategy, particularly when A is nearly symmetric in some sense, to the relation of LGMRES to the ORTHOMIN algorithm, which was mentioned in section 3.3. In fact, in our experiments we find that LGMRES performs best when k is much smaller than m (typically $k \leq 3$), whereas GCROT often prefers $k > m$. Additionally, as previously mentioned, the LGMRES(m, k) truncation strategy results in a more straightforward implementation that lends itself to a block method.

4. Experimental results. We demonstrate the potential of LGMRES by presenting experimental results from a variety of problems using implementations of LGMRES in both MATLAB and a locally modified version of PETSc (Argonne National Laboratory's Portable, Extensible Toolkit for Scientific Computation) [3, 4]. We tested problems from various sources, including the Matrix Market Collection [23] and the University of Florida Sparse Matrix Collection [6]. In sections 4.1 and 4.2, we compare MATLAB implementations of LGMRES(m, k), GMRES(m), GCRO [7], and GCROT [8] for problems without preconditioning. In section 4.3, we demonstrate the usefulness of LGMRES for larger problems with preconditioning with a PETSc implementation of LGMRES.

4.1. Comparison to GMRES(m). In this section, we demonstrate that LGMRES can significantly accelerate the convergence of restarted GMRES. To compare the performance of LGMRES(m, k) and GMRES(m), we implemented each in MATLAB. Our purpose with these implementations is to gauge the acceleration potential of LGMRES as well as its range of applicability on a variety of problems. Therefore, in this section and section 4.2, we do not use preconditioning for the MATLAB tests, allowing iteration counts to be large. A zero initial guess is used for all problems.

We look at a test set of 18 problems, 15 from the Matrix Market Collection and 3 convection-diffusion (CD) problems. The Matrix Market problems include the following: add20, orsreg_1, orsirr_1, cdde1, pde900, sherman1, sherman4, rdbl1250, cavity05, nos3, watt_2, fs_760_1, e05r0000, steam2, and cavity10. If a right-hand side is not provided, we generate a random right-hand side. The three CD problems are taken from [18] and are variations of the PDE $u_{xx} + u_{yy} + Du_x = -(41)^2$ with increasing degree of nonsymmetry: $D = 1$, $D = 41$, and $D = 41^2$, which we refer to as morgan_1, morgan_41, and morgan_1681, respectively. These PDEs are discretized by central finite differences on the unit square with zero boundary conditions and step-size $h = 1/41$. We stop the iteration when the relative residual norm is less than the convergence tolerance ζ, i.e., when $\|r_i\|_2/\|r_0\|_2 \leq \zeta$. We use $\zeta = 10^{-5}$ for all problems in this comparison. Several restart parameters are chosen for each problem, resulting in a total of 53 test cases. In particular, for the first 11 Matrix Market problems (in the preceding list) and the three CD problems, we use $m = 10$, 20, and 30. We use $m = 10$, 20 for problem fs_760_1 and $m = 20$, 30, and 40 for the last three Matrix Market problems.

For each of these 53 test cases, we compare the performances of GMRES(m) and LGMRES with equal-sized approximation spaces. Figure 3 shows the number of matrix-vector multiplies required for convergence for GMRES(m) and LGMRES($m - k$, k) with $k = 1 : 5$. In both the top and bottom plots, the y-axis is the number of matrix-vector multiplies required for convergence by GMRES(m) divided by the number required by LGMRES($m - k$, k). Note that the log of this ratio is plotted on the y-axis of Figure 3. The x-axis corresponds to the 15 Matrix Market problems followed by the three CD problems in the order given in the previous paragraph, and results for the same problem with increasing m are adjacent. For example, test case 1 corresponds to problem add20 with $m = 10$, for which GMRES(m) re-

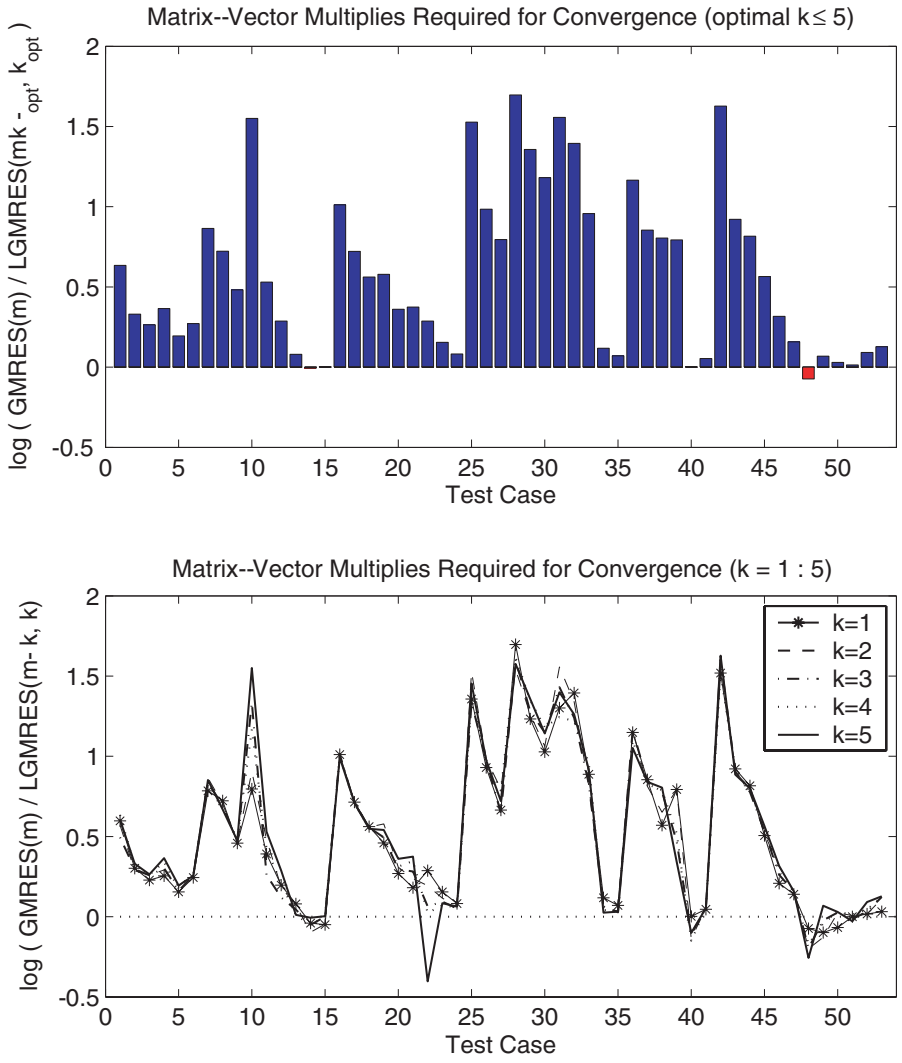


FIG. 3. *A comparison of the number of matrix-vector multiplies required for convergence by GMRES(m) and LGMRES(m − k, k) for* 53 *test cases. The top panel compares GMRES(m) to the "best" LGMRES(m, k). The bottom panel displays results for LGMRES(m − k, k) versus GMRES(m) for five different values of k* ($k = 1 : 5$).

quires approximately four times as many matrix-vector multiplies as does LGMRES $(m - k, k)$.

In the top panel of Figure 3, the result of the "best" LGMRES$(m - k, k)$ for $k = 1 : 5$ is compared to GMRES(m). The bars extending above the x-axis favor LGMRES$(m - k, k)$ (51 cases)—in these cases GMRES(m) requires more matrix-vector multiplies than does LGMRES$(m - k, k)$. The bars below the x-axis favor GMRES(m) (two cases: pde900 with $m = 20$ and morgan_41 with $m = 10$). Ratios of improvement (as opposed to iteration counts) are given to demonstrate the potential improvement with LGMRES, though we note that the number of iterations required by LGMRES$(m - k, k)$ is less than n (where n is the matrix order) in 46 of the 53

TABLE 2

Results for LGMRES(29, 1). Problem size, iterations required for $\|r_i\|_2/\|r_0\|_2 \leq 10^{-9}$, median skip angle, and median sequential angle are listed for each problem.

Problem	Size (n)	Iterations (matrix-vector multiplies)	Median seq. angle $\angle(r_i, r_{i-1})$	Median skip angle $\angle(r_{i+1}, r_{i-1})$
add20	2395	606 (587)	63.0	79.0
orsirr_1	1030	2190 (2118)	41.0	55.4
orsreg_1	2205	515 (499)	72.2	84.6
sherman_1	1000	757 (733)	61.7	76.4

test cases. In the remaining seven cases (steam2 with $m = 20$ and both e05r0000 and orsirr_1 with $m = 10$, 20, and 30), the number of iterations is less than $2.25n$. The number of iterations required by GMRES(m), on the other hand, is much greater than n for a number of these test cases, as reflected by several large ratios in the top panel of Figure 3.

The plot in the bottom panel of Figure 3 shows the variance in results for LGMRES($m-k$, k) with $k = 1 : 5$. Generally $k \leq 3$ is best for LGMRES($m-k$, k), and in our experiments, returns are diminishing for larger k, especially when m is small.

Furthermore, as with the majority of these test problems in Figure 3, we typically observe that the percentage improvement of LGMRES over GMRES decreases with increasing m. This trend is likely related to smaller values of m resulting in larger iteration counts and more noticeable alternating behavior.

Experimentally, we observe that LGMRES nearly always has a larger median skip angle than does GMRES(m). For example, in Table 2 we list the LGMRES(29, 1) results for the same four problems for which GMRES(30) results were provided in Table 1 in section 3.1. Again, the number of iterations required for convergence ($\|r_i\|_2/\|r_0\|_2 \leq 10^{-9}$) as well as the median sequential and median skip angle values are listed.

Consider two consecutive approximation spaces $\mathcal{S}_i$ and $\mathcal{S}_{i+1}$. As compared to standard GMRES(m), LGMRES(m, k) does not necessarily affect how much of $\mathcal{S}_{i+1}$ can be found in $\mathcal{S}_i$. However, it does typically "improve orthogonality" quite significantly between the current approximation space and the space generated two restart cycles ago: $\mathcal{S}_{i+1}$ and $\mathcal{S}_{i-1}$. This action accelerates the convergence over that of GMRES(m) in many cases. Recall from Theorem 4 that the size of the sequential angles is directly related to the reduction in residual at each cycle. Therefore, if increasing the skip angles occurs at the expense of reducing the average sequential angle, then LGMRES augmenting slows convergence. Intuitively, the method that "wins" generally has large skip angles *and* large sequential angles.

Our experiments seem to indicate that the LGMRES augmenting scheme significantly improves GMRES(m) convergence under the following conditions: GMRES(m) skip angles are small and continue to decrease as the iteration progresses; GMRES(m) sequential angles are relatively small and converging to the same angle as the iteration progresses; or the average skip angle increases significantly after LGMRES augmenting. All of these conditions are typically met for problems that display alternating behavior, although some or all are evident in other problems as well. On the other hand, LGMRES is not as helpful when one of the following occurs: GMRES(m) skip angles are not small; GMRES(m) sequential angles vary greatly from cycle to cycle; GMRES(m) converges in a small number of iterations; or GMRES(m) skip angles and sequential angles are near zero, indicating stalling. We believe that the LGM-

TABLE 3

A comparison of matrix-vector multiplies required for convergence ($\|r_i\|_2/\|r_0\|_2 \leq 10^{-9}$) for $u_{xx} + u_{yy} + Du_x = -(41)^2$, discretized by centered finite differences on the unit square with zero boundary conditions and step-size $h = 1/41$.

Matrix	D	$\frac{\|A-A^T\|_2}{\|A\|_2}$	m	GMRES(m)	LGMRES(m,1)	Adaptive
morgan_1	1	.005	10	735	245	245
			20	415	260	260
			30	272	199	199
morgan_41	41	.22	10	168	252	169
			20	200	301	301
			30	236	296	236
morgan_1681	41^2	.99	10	496	475	464
			20	486	453	469
			30	488	482	477

RES augmenting scheme most benefits problems that are close to symmetric in some sense as these are the problems for which alternating is most easily explained, but we have seen the algorithm perform well for a variety of problems.

Though we have found that scalar measurements of symmetry generally do not correlate with LGMRES performance, a close look at the three aforementioned CD problems with increasing degree of nonsymmetry does provide some insight into LGMRES convergence behavior. In Table 3, results similar to those presented in Figure 3 are listed. However, now we compare GMRES(m) with LGMRES(m, 1) to better examine the effect of appending one error approximation to the Krylov subspace. Whereas previously presented results compared methods with equal-sized approximation spaces or equal storage requirements, here the methods have equal-sized Krylov subspaces at each cycle.

For morgan_1, the coefficient matrix A is nearly symmetric. LGMRES(m, 1) is effective for this problem, particularly for the $m = 10$ case where the GMRES(m) residual vectors alternate noticeably. (The median skip angles in degrees for GMRES(m) are .6, 2.4, and 23.4 for $m = 10$, 20, and 30, respectively.) On the other hand, morgan_41 with $D = 41$ is an excellent example of the type of problem for which LGMRES performs very poorly. Because this problem converges fairly quickly with GMRES(m) and is far from symmetric, we did not expect LGMRES(m, 1) to be very helpful. But the fact that LGMRES(m, 1) actually slows convergence considerably was unanticipated. However, we have since observed that LGMRES generally performs poorly on problems for which the GMRES(m) iteration count increases with increasing m, such as morgan_41. Finally, morgan_1681 is nearly skew-symmetric and benefits only slightly from the augmenting scheme of LGMRES(m, 1). In general, we find in our experiments that nearly skew-symmetric problems do not benefit as much from LGMRES as do nearly symmetric problems.

The morgan_41 problem highlights the need for a potential improvement to the LGMRES algorithm; in particular, an adaptive version that determines whether or not to augment would be beneficial. Designing a simple adaptive LGMRES algorithm effective for all test cases and for all values of m has proved difficult. Our most promising effort to date is described at the end of section 3.3. Results for this algorithm are given in the right column of Table 3 and are decidedly mixed. While this adaptive method usually mitigates the extent to which LGMRES fails on tricky problems, it can be less effective than standard LGMRES on others. Deciding whether or not to

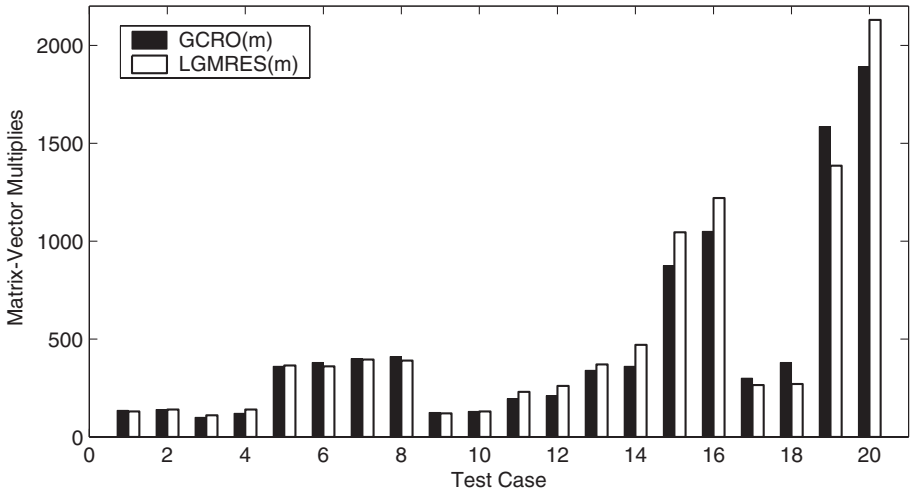


FIG. 4. *A comparison of the numbers of matrix-vector multiplies required for convergence by nontruncated GCRO and full LGMRES. Test cases* 1–20 *correspond to results for* $m = 5$ *followed by* $m = 10$ *for morgan_1, morgan_41, morgan_1681, sherman1, sherman4, add20, cavity05, orsirr_1, orsreg_1, and pores_3.*

augment within a given restart cycle is difficult. We find that analysis within a single restart cycle is not sufficient as augmenting has a cumulative effect. An analysis of convergence across cycles (for both GMRES and LGMRES) would provide a better understanding of the behavior of LGMRES and enable us to design a more effective adaptive strategy.

The effectiveness of LGMRES depends upon the matrix problem and the restart parameter m, but the savings in matrix-vector multiplies are quite substantial in many cases. Though many of the test problems presented in this section would benefit from preconditioning, results for problems such as cavity10 that are difficult to precondition [23] are encouraging. And, in our experience, we find that LGMRES typically does not require more iterations than does restarted GMRES.

4.2. Comparison to GCRO and GCROT. In section 3.4, we discuss the similarities (and differences) between LGMRES(m, k) and GCROT. Here we compare the performance of the two truncated methods, first briefly examining their less practical nontruncated counterparts, full GCRO and full LGMRES (LGMRES(m)).

We evaluate MATLAB implementations of LGMRES(m) and GCRO in the same manner as in section 4.1. That is, we compare the number of matrix-vector multiplies required for the relative residual norm to be less than the convergence tolerance ζ. For these nontruncated methods, we use small values of m, $m = 5$ and $m = 10$, since storage increases with each iteration. We again test the three related CD problems (morgan_1, morgan_41, and morgan_1681) with $\zeta = 10^{-9}$, as these problems were also used in [8]. In addition, we compare results for a subset of the Matrix Market problems from the previous section with $\zeta = 10^{-5}$ as well as one new Matrix Market problem, pores_3, that stalls for both GMRES(10) and GMRES(5).

Figure 4 compares the two methods and indicates that the performance of the two methods, in terms of matrix-vector multiplies, is often similar. In terms of convergence, our experiments seem to indicate that appending vectors to the end of the standard Krylov subspace (as LGMRES does) is not necessarily inferior to orthogo-

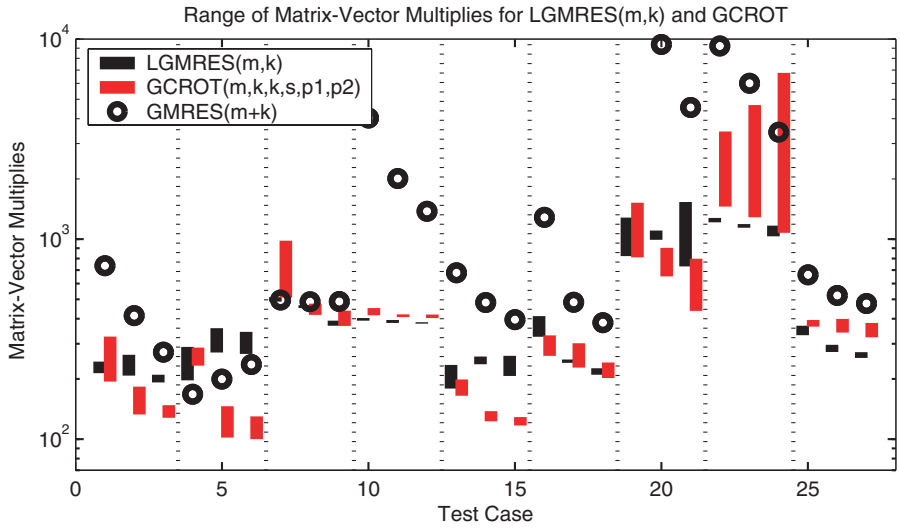


FIG. 5. *A comparison of the minimum to maximum number of matrix-vector multiplies required for convergence by LGMRES(m, k) and GCROT(m, k, k, s, p1, p2) for constant-sized approximation spaces. GMRES(m+k) is also indicated. Test cases* 1–27 *correspond to* $m+k = 10$, $m+k = 20$, *and* $m + k = 30$, *respectively, for morgan_1, morgan_41, morgan_1681, sherman1, sherman4, add20, cavity05, orsirr_1, and orsreg_1.*

nalizing against them at the start of the cycle, and our experience does not clearly indicate which approach is to be preferred in a given situation. Even in the case where GMRES(m) stalls (and intuitively LGMRES(m) would not be helpful), one can find counterexamples such as pores_3 (cases 19 and 20) where slow initial convergence is eventually overcome.

As mentioned in section 3.3, though interesting from a theoretical point of view, nontruncated methods are often impractical due to increasing storage requirements. For example, Figure 4 indicates that both orsirr_1 and pores_3 require storing more than n vectors. Additionally, for LGMRES(m), an increasing number of orthogonalizations are required in each cycle. Therefore, we do not further investigate the convergence behavior of LGMRES(m) but instead focus on a comparison of the more practical versions of the two algorithms: LGMRES(m, k) and GCROT(m, k, k, s, $p1$, $p2$).

For each of these truncated algorithms, the size of the approximation space is $m + k$. We use a MATLAB implementation of GCROT supplied by Oliver Ernst. The test problems are the same as in Figure 4 with approximation spaces of size 10, 20, and 30, although the pores_3 test cases have been dropped since neither truncated algorithm converges for that problem.

For each of the 20 test cases, we ran LGMRES(m, k) with $k = 1 : 3$, as this range was recommended in the previous section. Additionally, ten permutations of GCROT(m, k, k, s, $p1$, $p2$), where $m + k$ is constant, were chosen to reflect the choices in [8] (e.g., $m \leq k$, $s \leq \lceil \frac{m}{2} \rceil$). Figure 5 compares the two methods for all 27 test cases. The bars indicate the range (minimum to maximum) of matrix-vector multiplies required. The circles represent restarted GMRES for each problem with the corresponding approximation space size. Vertical dotted lines separate test cases corresponding to the same matrix problem. The missing circle for test case 19 in-

dicates that GMRES($m + k$) required more than 10,000 iterations. Some of these iteration counts are unrealistically large, but recall that we are not considering preconditioning and are simply evaluating the relative performance of the two algorithms.

Results for the two algorithms are relatively similar in most of the test cases. We again notice that LGMRES(m, k) has particular difficulty with morgan_41 (GCROT has difficulty only for $m + k = 10$). Problem morgan_1681 is somewhat resistant to improvement by both methods, and problem orsirr_1 is highly sensitive to the choice of input parameters with GCROT. It is not clear in our experience or from results presented in [8] how to choose the optimal parameters for GCROT. For the ten of the many possible permutations we chose for GCROT for each fixed $m + k$, there was no observable trend as to which of the ten permutations were most (or least) effective across this set of test problems. In addition, we have found that occasionally $m > k$ can be more effective than the recommended $k > m$ for GCROT (in test cases for problems add20 and orsirr_1, for example). Ernst also found that choosing the parameters for GCROT can be problematic [14]. However, for LGMRES, $k \leq 3$ is nearly always the best choice and the variation in results for $k = 1 : 3$ is generally reasonable.

4.3. Effectiveness for larger preconditioned problems. In this section, we demonstrate that LGMRES can be a helpful addition to preconditioning. We implemented LGMRES in C using a locally modified version of PETSc 2.1.5 [3, 4] in order to easily access preconditioners, test larger problems, and obtain reliable timing results (instead of counting matrix-vector multiplies). Our PETSc implementation is available in PETSc 2.1.6. First, we look at cumulative results for 15 different matrix problems. Then we more closely examine a few of those problems.

We chose a variety of test problems from the Matrix Market Collection [23], the University of Florida (UF) Sparse Matrix Collection [6], and the PETSc [3, 4] collection of test matrices. We use the ILU(p) preconditioner, where p indicates the level of fill (see, e.g., [25]). If a right-hand side is not provided, we generate a random right-hand side. For reference, the test problems are listed in Table 4.

TABLE 4

List of test problems together with the matrix order (n), number of nonzeros (nnz), preconditioner, source, and description of the application area (if known). Source indicates Matrix Market Collection (MM), UF Sparse Matrix Collection (UF), or PETSc test collection (PC), along with a set or directory name if applicable.

	Problem	n	nnz	ILU(p)	Source	Application area
1	fidapm11	22294	623554	ILU(0)	MM: Sparskit	fluid flow
2	memplus	17758	126150	ILU(0)	MM: Hamm	digital circuit simulation
3	arco3	38194	241066	ILU(0)	PC	multiphase flow: oil reservoir
4	arco5	35388	154166	ILU(0)	PC	multiphase flow: oil reservoir
5	arco6	108009	2204937	ILU(0)	PC	multiphase flow: oil reservoir
6	ex40	7740	458012	ILU(0)	UF: FIDAP	fluid flow
7	garon2	13535	390607	ILU(1)	UF: Garon	fluid flow
8	bcircuit	68902	375558	ILU(1)	UF: Hamm	digital circuit simulation
9	xenon1	48600	1181120	ILU(2)	UF: Ronis	crystalline compound analysis
10	pesa	11738	79566	ILU(0)	UF: Gaertner	
11	aft01	8202	125567	ILU(0)	UF: Okunbor	acoustic radiation
12	venkat50	62424	1717792	ILU(0)	UF: Simon	fluid dynamics
13	epb3	84617	463625	ILU(0)	UF: Averous	heat exchanger simulation
14	big	13209	91465	ILU(1)	UF: Gaertner	
15	zhao2	33861	166453	ILU(0)	UF: Zhao	electromagnetic systems

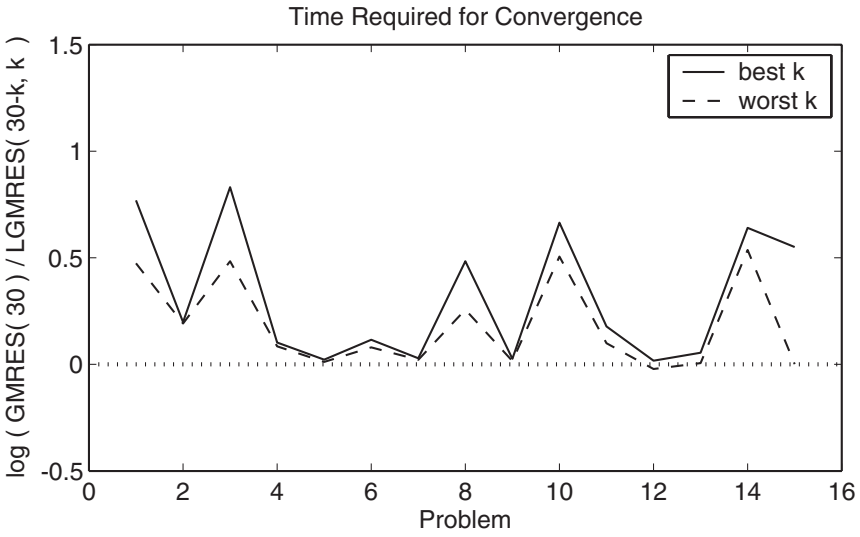


FIG. 6. *A comparison of the time required for convergence for* 15 *different problems with GMRES(*30*) and LGMRES(*$30-k$, k*), $k = 1:3$. All methods use an approximation space of dimension* 30.

We compare the performance of restarted GMRES to that of LGMRES(m, k) with the same approximation space size and then the same storage requirements. For LGMRES, we report results for $k = 1:3$, as we find that choosing k in this range typically results in the most improvement with the least risk of increasing execution time. All tests are run until the relative residual norm is less than the convergence tolerance $\zeta = 10^{-9}$. Recall that GMRES with left preconditioning minimizes the preconditioned residual norm ($\|M^{-1}r\|_2$), and, therefore, the determination of convergence is based on this preconditioned residual norm as usual. The initial guess is a zero vector in all cases. Unless otherwise noted, results provided were run on a Sun UltraSPARC 10 with 256M RAM, a clock-rate of 360 MHz, a 16KB L1 cache, and a 2MB L2 cache. For each problem we report wall clock time for the linear solve. All timings are averages from five runs and have standard deviations of at most two percent, although most are less than one percent. If a method does not converge in 30000 iterations, the execution time reported reflects the time to 30000 iterations, and we say that the method does not converge. Note that iteration counts for problems that converge are well below 30000. We did not compare LGMRES(m, k) to GCROT for these problems because no PETSc implementation of GCROT is available.

In Figure 6, we compare GMRES(30) to LGMRES(29, 1), LGMRES(28, 2), and LGMRES(27, 3). All four of these methods generate an approximation space of dimension 30 during each restart cycle. Similar to the plots seen previously, the y-axis indicates the log of the ratio of the time to converge for GMRES(30) to the time to converge for both the best and worst performing cases of LGMRES($30-k$, k) for $k = 1:3$, and the x-axis corresponds to the 15 test problems in Table 4. Points above the x-axis favor LGMRES and points below favor GMRES. Note that GMRES(30) does not converge (in 30000 iterations) for problems 10, 14, and 15, and LGMRES(27, 3) does not converge for problem 15.

For larger problems in particular, comparing restarted GMRES to an LGMRES method that requires an equal amount of storage is also of interest. Both

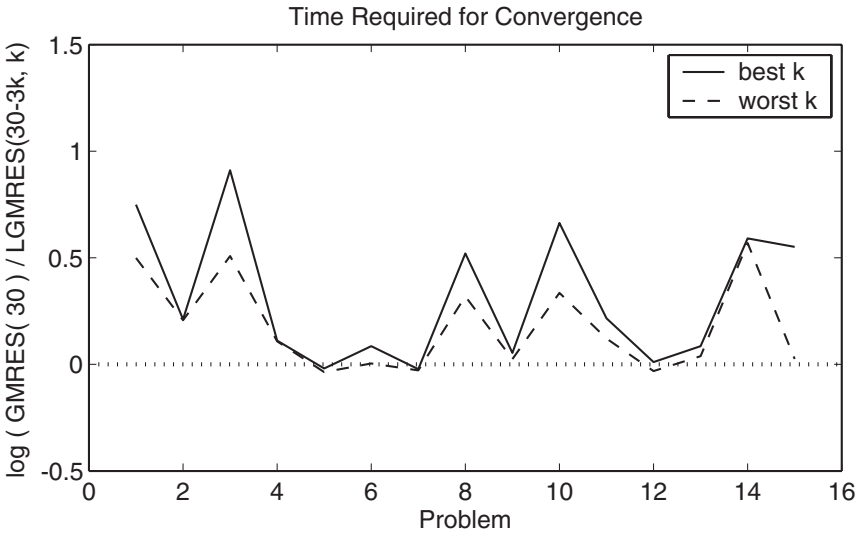


FIG. 7. *A comparison of the time required for convergence for* 15 *different problems with GMRES(*30*) and LGMRES(*$30-3k$*,* k*) with* $k = 1:3$*. All methods require storage for* 33 *vectors of length* n*.*

GMRES(30) and LGMRES($30-3k$, k) have the same 33 vector storage requirement (see section 3.2). Similar to the previous figure, Figure 7 compares GMRES(30) to LGMRES(27, 1), LGMRES(24, 2), and LGMRES(21, 3). In this comparison, one augmentation vector must be more helpful than three standard Krylov vectors for LGMRES to win. This requirement is fairly stringent for some of the larger problems given that we allow only 33 vectors of storage. Nevertheless, the majority of the problems still show improvement with LGMRES.

Now we examine problems bcircuit, fidapm11, and big from our test set (in Table 4) in more detail, additionally providing timing results for full GMRES. The results in Tables 5–7 for these three problems demonstrate different possible relations in convergence behavior among LGMRES(m, k), GMRES(m), and full GMRES.

First consider the timing results for problem bcircuit in Table 5. For this problem, full GMRES requires memory resources beyond the physical memory limit of our

TABLE 5

Results for matrix bcircuit and its corresponding right-hand side, with $n = 68902$*,* $nnz = 375558$*, and ILU(*1*) preconditioning. Times are in seconds and include mean and standard deviations of times for five runs.*

Method	Approx. space dimension	# vectors stored	Matrix-vector multiplies	Time
Full GMRES	1013	1016	1013	2880.364 ± 9.24
GMRES(30)	30	33	5602	1135.38 ± 12.58
LGMRES(29,1)	30	35	2959	615.28 ± 5.61
LGMRES(28,2)	30	37	1730	365.16 ± 2.47
LGMRES(27,3)	30	39	1707	369.70 ± 2.48
LGMRES(27,1)	28	33	2631	533.42 ± 4.54
LGMRES(24,2)	26	33	2467	503.71 ± 3.54
LGMRES(21,3)	24	33	1672	339.42 ± 2.21

TABLE 6

*Results for matrix fidapm*11 *and its corresponding right-hand side, with* $n = 22294$, $nnz = 623554$, *and ILU(*0*) preconditioning. Times are in seconds and include mean and standard deviations of times for five runs.*

Method	Approx. space dimension	# vectors stored	Matrix-vector multiplies	Execution time
Full GMRES	952	955	952	854.02 ± 6.27
GMRES(30)	30	33	16482	2100.23 ± 8.40
LGMRES(29,1)	30	35	5511	704.65 ± 0.18
LGMRES(28,2)	30	37	2915	376.64 ± 0.10
LGMRES(27,3)	30	39	2733	357.19 ± 0.78
LGMRES(27,1)	28	33	5239	664.84 ± 1.89
LGMRES(24,2)	26	33	3399	431.39 ± 1.00
LGMRES(21,3)	24	33	2941	373.96 ± 0.41

TABLE 7

Results for matrix big with a random right-hand side, with $n = 13209$, $nnz = 91465$, *and ILU(*1*) preconditioning. Times are in seconds and include mean and standard deviations of times for five runs.*

Method	Approx. space dimension	# vectors stored	Matrix-vector multiplies	Execution time
Full GMRES	188	191	188	21.70 ± 0.03
GMRES(30)	30	33	> 30000	1231.54 ± 1.08
LGMRES(29,1)	30	35	8500	358.07 ± 0.15
LGMRES(28,2)	30	37	6997	300.18 ± 0.32
LGMRES(27,3)	30	39	6546	281.79 ± 0.70
LGMRES(27,1)	28	33	7654	315.7 ± 0.37
LGMRES(24,2)	26	33	7971	327.96 ± 0.20
LGMRES(21,3)	24	33	8259	332.56 ± 0.09

machine. For this reason, we had to rerun the bcircuit problem on a similar machine with four times as much memory (a Sun UltraSPARC 10 with 1024M RAM, a clock-rate of 440 MHz, a 16KB L1 cache, and a 2MB L2 cache) to obtain timing results for full GMRES. Therefore, for Table 5 only, all results presented for bcircuit were obtained on this second machine. Even with the extra memory provided by the second machine, we see that restarted GMRES(30) is more than twice as fast as full GMRES, and LGMRES is even faster. Conversely, for problem fidapm11 given in Table 6, full GMRES is faster than GMRES(30) on our machine, although LGMRES still has the faster execution time of the three methods on this problem. Finally, results for problem big are given in Table 7. This third problem is interesting because GMRES(30) converges very slowly. In fact, the relative residual norm is still $\approx .002$ after 30000 iterations. Both LGMRES($30 - 3k$, k) and LGMRES($30 - k$, k), on the other hand, improve convergence dramatically over that of GMRES(30). However, for this moderately sized problem, full GMRES requires only 188 iterations and wins by a landslide.

Most of the problems presented here require a restarted method given the resources of the machine chosen for the experiments. On a more powerful machine (more memory and faster processor), full GMRES might be faster for many of these problems. At the same time, because every machine has a limit as to the size problems it can reasonably solve with a full method, restarted methods and acceleration methods provide a great advantage.

Finally, we note that because LGMRES is an accelerator, it is not, in general, a substitute for an effective preconditioner. Although we did encounter a number of test problems for which the ILU preconditioner is not a viable option and LGMRES is a dramatic improvement over GMRES(m), we expect that in those cases an appropriate preconditioner would be even more effective. Nevertheless, LGMRES can be an effective addition to preconditioning for a range of problems. Although LGMRES improvements with preconditioning tend not to be as spectacular as the improvements seen for the nonpreconditioned problems of section 4.1, even moderate acceleration for large problems can translate into significant time savings.

5. Concluding remarks. In this paper, we have described a method that accelerates the convergence of GMRES(m). We have also discussed some interesting observed properties of the convergence of GMRES(m) that motivated the algorithm's development. Experimental results demonstrate that the LGMRES augmentation scheme is an effective accelerator for GMRES(m) with or without preconditioning. Furthermore, the algorithm is straightforward and easy to implement. However, LGMRES is not typically a substitute for preconditioning and does not help when a problem stalls for a given restart parameter. Possible improvements to the algorithm include a robust adaptive variant. In future work, we will describe a more memory-efficient block implementation of the LGMRES algorithm.

Acknowledgments. We thank Oliver Ernst for providing us with his MATLAB implementation of GCROT and the referees for their many helpful suggestions.

REFERENCES

[1] S. F. Ashby, T. A. Manteuffel, and P. E. Saylor, *A taxonomy for conjugate gradient methods*, SIAM J. Numer. Anal., 27 (1990), pp. 1542–1568.

[2] A. Baker, J. Dennis, and E. R. Jessup, *Toward memory-efficient linear solvers*, in VECPAR'2002, Fifth International Conference on High Performance Computing for Computational Science: Selected Papers and Invited Talks, J. Palma, J. Dongarra, V. Hernandez, and A. A. Sousa, eds., Lecture Notes in Comput. Sci. 2565, Springer-Verlag, New York, 2003, pp. 315–327.

[3] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, L. C. McInnes, and B. F. Smith, *PETSc home page*, http://www.mcs.anl.gov/petsc, 2001.

[4] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *PETSc Users Manual*, Tech. report ANL-95/11, Revision 2.1.3, Argonne National Laboratory, Argonne, IL, 2002.

[5] A. Chapman and Y. Saad, *Deflated and augmented Krylov subspace techniques*, Numer. Linear Algebra Appl., 4 (1997), pp. 43–66.

[6] T. Davis, *University of Florida sparse matrix collection*, http://www.cise.ufl.edu/research/sparse/matrices, 2002.

[7] E. de Sturler, *Nested Krylov methods based on GCR*, J. Comput. Appl. Math., 67 (1996), pp. 15–41.

[8] E. de Sturler, *Truncation strategies for optimal Krylov subspace methods*, SIAM J. Numer. Anal., 36 (1999), pp. 864–889.

[9] E. de Sturler and D. R. Fokkema, *Nested Krylov methods and preserving orthogonality*, in Sixth Copper Mountain Conference on Multigrid Methods, N. Melson, T. Manteuffel, and S. McCormick, eds., Part 1 of NASA Conference Publication 3324, NASA, 1993, pp. 111–126.

[10] M. Eiermann and O. G. Ernst, *Geometric aspects in the theory of Krylov subspace methods*, in Acta Numerica, Acta Numer. 10, Cambridge University Press, Cambridge, UK, 2001, pp. 251–312.

[11] M. Eiermann, O. G. Ernst, and O. Schneider, *Analysis of acceleration strategies for restarted minimum residual methods*, J. Comput. Appl. Math., 123 (2000), pp. 261–292.

[12] S. C. Eisenstat, H. C. Elman, and M. H. Schultz, *Variational iterative methods for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal., 20 (1983), pp. 345–357.

[13] M. Embree, *The tortoise and the hare restart GMRES*, SIAM Rev., 45 (2003), pp. 259–266.

[14] O. Ernst, *A numerical study of acceleration schemes for restarted minimum residual methods*, Presentation at the Seventh Copper Mountain Conference on Iterative Methods, Copper Mountain, CO, 2002.
[15] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, *Accelerated inexact Newton schemes for large systems of nonlinear equations*, SIAM J. Sci. Comput., 19 (1998), pp. 657–674.
[16] A. Greenbaum and Z. Strakoš, *Matrices that generate the same Krylov residual spaces*, in Recent Advances in Iterative Methods, G. Golub, A. Greenbaum, and M. Luskin, eds., Springer-Verlag, New York, 1994, pp. 95–118.
[17] W. Joubert, *On the convergence behavior of the restarted GMRES algorithm for solving nonsymmetric linear systems*, Numer. Linear Algebra Appl., 1 (1994), pp. 427–447.
[18] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.
[19] R. B. Morgan, *Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1112–1135.
[20] R. B. Morgan, *GMRES with deflated restarting*, SIAM J. Sci. Comput., 24 (2002), pp. 20–37.
[21] N. M. Nachitgal, L. Reichel, and L. N. Trefethen, *A hybrid GMRES algorithm for nonsymmetric linear systems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 796–825.
[22] N. M. Nachtigal, S. C. Reddy, and L. N. Trefethen, *How fast are nonsymmetric matrix iterations?*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 778–795.
[23] National Institute of Standards and Technology, Mathematical and Computational Sciences Division, *Matrix Market*, http://math.nist.gov/MatrixMarket, 2002.
[24] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.
[25] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.
[26] Y. Saad, *Analysis of augmented Krylov subspace methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 435–449.
[27] 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.
[28] V. Simoncini, *On the convergence of restarted Krylov subspace methods*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 430–452.
[29] H. A. van der Vorst and C. Vuik, *The superlinear convergence behavior of GMRES*, J. Comput. Appl. Math., 48 (1993), pp. 327–341.
[30] H. A. van der Vorst and C. Vuik, *GMRESR: A family of nested GMRES methods*, Numer. Linear Algebra Appl., 1 (1994), pp. 369–386.
[31] D. M. Young and K. C. Jea, *Generalized conjugate-gradient acceleration of nonsymmetrizable iterative methods*, Linear Algebra Appl., 34 (1980), pp. 154–194.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 4, pp. 985–1000

NUMERICAL STABILITY OF THE PARALLEL JACOBI METHOD*

T. LONDRE† AND N. H. RHEE‡

Abstract. In this paper we study numerical stability of the parallel Jacobi method for computing the singular values and singular subspaces of an invertible upper triangular matrix that is obtained from QR decomposition with column pivoting. We show that in this case the parallel Jacobi method locates singular values and singular subspaces to full machine accuracy.

Key words. roundoff error, perturbation theory, parallel Jacobi method, singular values, singular subspaces

AMS subject classifications. 65F99, 65G05, 15A18

DOI. 10.1137/S0895479802415995

1. Introduction.

Mathias [5] has observed that the reduction of a rectangular matrix of full column rank to a smaller invertible upper triangular matrix R using QR decomposition is stable. In this paper we study numerical stability of the parallel Jacobi method for computing the singular values and singular subspaces of an invertible upper triangular matrix obtained from QR decomposition with column pivoting. It turns out that in this case the parallel Jacobi method locates singular values and singular subspaces to full machine accuracy. This means that the error introduced is dominated by error from the QR part of the process. This paper is organized as follows. In section 2 we recall some perturbation theory of singular values and singular subspaces. In section 3 we present stable angle formulas for SVD of upper triangular matrices based on [1]. In section 4 we describe the mobile parallel Jacobi method (MPJM). In section 5 roundoff error for this method is analyzed. Finally in section 6 we give some numerical results. In this paper, we use MATLAB notation freely.

2. Perturbation theory.

In this section we present two basic theorems which will be used to prove the stability of the parallel Jacobi method. We first state a perturbation theorem [2] for singular values.

THEOREM 1. *Let G and $\tilde{G} = G + \triangle G$ be $n \times n$ real invertible matrices such that $\eta \equiv \|G^{-1}(\triangle G)\|_2 < 1$. Then*

$$\frac{|\tilde{\sigma}_i - \sigma_i|}{\sigma_i} \leq \eta,$$

where σ_i and $\tilde{\sigma}_i$ are the ith largest singular values of G and $\tilde{G}$, respectively.

Now we state a theorem which can be used to find a perturbation bound for left singular subspaces of invertible square matrices.

THEOREM 2. *Let G and $\tilde{G} = G + \triangle G$ be $n \times n$ real invertible matrices such that $\eta \equiv \|G^{-1}(\triangle G)\|_2 < 1/3$. Let $G = U\Sigma V^T$ and $\tilde{G} = \tilde{U}\tilde{\Sigma}\tilde{V}^T$ be singular value decompositions where σ_i and $\tilde{\sigma}_i$ are the ith largest singular values of G and $\tilde{G}$, respectively,*

*Received by the editors October 10, 2002; accepted for publication (in revised form) by I. C. F. Ipsen July 7, 2004; published electronically May 6, 2005.
http://www.siam.org/journals/simax/26-4/41599.html
†Department of Mathematics, Blue River Community College, Independence, MO 64057 (londret@blueriver.cc.mo.us).
‡Department of Mathematics and Statistics, University of Missouri at Kansas City, Kansas City, MO (rheen@umkc.edu).

and also the ith diagonal entries of Σ *and* $\tilde{\Sigma}$. *If* $S = U^T\tilde{U}$, *then*

$$|s_{ij}| \leq \frac{3\eta}{\sqrt{1-3\eta}} \frac{\sigma_i\ \tilde{\sigma}_j}{\left|\sigma_i^2 - \tilde{\sigma}_j^2\right|},$$

provided $\sigma_i \neq \tilde{\sigma}_j$.

Proof. Let $H = GG^T$ and $\tilde{H} = \tilde{G}\tilde{G}^T$. If we write $\tilde{H} = H + \triangle H$, then $\triangle H = G(\triangle G)^T + (\triangle G)G^T + (\triangle G)(\triangle G)^T$. Note that $\eta = \|\Sigma^{-1}U^T(\triangle G)\|_2$. Since $H^{-1/2} = U\Sigma^{-1}U^T$, it follows that

$$\left\|H^{-1/2}(\triangle H)H^{-1/2}\right\|_2 \leq 2\eta + \eta^2 \leq 3\eta < 1.$$

Since σ_i^2, U and $\tilde{\sigma}_j^2$, $\tilde{U}$ are the eigenvalues and eigenvector matrices of positive definite matrices H and $\tilde{H}$, respectively, the theorem follows from Theorem 1 of [4]. ∎

Remark 1. To see how we can use Theorem 2, recall that if U and $\tilde{U}$ are $n \times n$ orthogonal matrices, and we partition $U = [U_a\ U_b]$ and $\tilde{U} = [\tilde{U}_a\ \tilde{U}_b]$, where U_a, $\tilde{U}_a$ and U_b, $\tilde{U}_b$ are $n \times k$ and $n \times (n-k)$ matrices, respectively, then the sines of the principal angles between the column spaces of U_a and $\tilde{U}_a$ are the singular values of the matrix $U_a^T\tilde{U}_b$. So we let $\mathrm{s}(U_a, \tilde{U}_a) \equiv \|U_a^T\tilde{U}_b\|_2$, which is the maximum sine of the principal angles between the column spaces of U_a and $\tilde{U}_a$.

Since

$$\mathrm{s}(U_a, \tilde{U}_a) = \left\|U_a^T\tilde{U}_b\right\|_2 \leq \left\|U_a^T\tilde{U}_b\right\|_F \leq \sqrt{k(n-k)} \max_{1\leq i\leq k,\ k+1\leq j\leq n} |s_{ij}|,$$

it follows from Theorem 2 and the fact that $\sqrt{k(n-k)} \leq n/2$ that

$$\mathrm{s}(U_a, \tilde{U}_a) \leq 1.5n\ \frac{\eta}{\sqrt{1-3\eta}} \max_{1\leq i\leq k,\ k+1\leq j\leq n} \frac{\sigma_i\ \tilde{\sigma}_j}{\left|\sigma_i^2 - \tilde{\sigma}_j^2\right|}.$$

Because of Theorem 1, the above inequality can be written

$$\mathrm{s}(U_a, \tilde{U}_a) \leq 1.5n\ \frac{\eta}{\mathrm{sep}(\Sigma_a, \Sigma_b)} + O(\eta^2), \tag{1}$$

where

$$\mathrm{sep}(\Sigma_a, \Sigma_b) \equiv \min_{1\leq i\leq k,\ k+1\leq j\leq n} \frac{\left|\sigma_i^2 - \sigma_j^2\right|}{\sigma_i\ \sigma_j}. \tag{2}$$

Remark 2. Note that the estimate for $\mathrm{s}(U_a, \tilde{U}_a)$ holds for the SVD of the form $G = (UP)(P^T\Sigma P)(VP)^T$, where P is any permutation. So, if P is chosen in such a way that U_a corresponds to (simple, multiple, or clustered) σ_i, the result (1) gives an estimate of how much the left singular subspace belonging to σ_i is perturbed. In this case $\mathrm{sep}(\Sigma_a, \Sigma_b)$ becomes the relative gap for σ_i in the set of singular values.

3. Stable angle formulas. In this section we present a set of angle formulas which is crucial for the stability of our parallel Jacobi method for upper triangular matrices. We also present some lemmas that will be used for error analysis.

The basic problem is the accurate calculation of the SVD of a 2×2 diagonal block

$$C = \begin{bmatrix} a & b \\ 0 & d \end{bmatrix}$$

of the given $n \times n$ invertible upper triangular matrix R. We use 2×2 reflectors

$$H_l = \begin{bmatrix} c_l & s_l \\ s_l & -c_l \end{bmatrix} \quad \text{and} \quad H_r = \begin{bmatrix} c_r & s_r \\ s_r & -c_r \end{bmatrix}.$$

In particular, we want $H_l^T C H_r = C_+$ to be diagonal. In fact, the following stable angle formulas are given according to those in [1], although the authors of [1] use

$$\begin{bmatrix} s & c \\ -c & s \end{bmatrix} \quad \text{instead of} \quad \begin{bmatrix} c & s \\ s & -c \end{bmatrix}$$

for diagonalization, which yields slight variations for $r,\ \zeta_l,\ t_r$, and $r,\ \zeta_r,\ t_l$ in the following formulas.

Case 1. $|a| \geq |d|$. We compute c_l, s_l, c_r, and s_r as follows:

1. $r = \frac{(a+d)(a-d)}{b}$,
2. $\zeta_l = \frac{r+b}{2d}$,
3. $t_l = \frac{1}{\zeta_l + \operatorname{sign}(\zeta_l)\sqrt{1+\zeta_l^2}}$,
4. $c_l = \frac{1}{\sqrt{1+t_l^2}}, \quad s_l = t_l c_l$,
5. $t_r = \frac{dt_l+b}{a}$,
6. $c_r = \frac{1}{\sqrt{1+t_r^2}}, \quad s_r = t_r c_r$.

Case 2. $|a| < |d|$. We compute c_r, s_r, c_l, and s_l as follows:

1. $r = \frac{(a+d)(a-d)}{b}$,
2. $\zeta_r = \frac{r-b}{2a}$,
3. $t_r = \frac{1}{\zeta_r + \operatorname{sign}(\zeta_r)\sqrt{1+\zeta_r^2}}$,
4. $c_r = \frac{1}{\sqrt{1+t_r^2}}, \quad s_r = t_r c_r$,
5. $t_l = \frac{at_r-b}{d}$,
6. $c_l = \frac{1}{\sqrt{1+t_l^2}}, \quad s_l = t_l c_l$.

We update the diagonal matrix $C_+ = \operatorname{diag}(a_+, d_+)$ by

$$a_+ = \frac{c_l}{c_r} a, \quad d_+ = \frac{c_r}{c_l} d. \tag{3}$$

We use the model of floating point number arithmetic with machine precision ε_M:

$$\operatorname{fl}(x * y) = x * y(1 + \varepsilon_1),$$

where "$*$" denotes addition, subtraction, multiplication, or division, and

$$\operatorname{fl}(\sqrt{x}) = \sqrt{x}(1 + \varepsilon_2),$$

where $|\varepsilon_i| \leq \varepsilon_M$ for $i = 1, 2$.

We state, without proof, two lemmas which will be used in obtaining bounds for errors in computed versions of c_l, c_r, s_l, and s_r.

LEMMA 3. *If x and y have the same sign,*

$$x(1 + \alpha_1) + y(1 + \alpha_2) = (x + y)(1 + \alpha),$$

where $|\alpha| \leq \max\{|\alpha_1|, |\alpha_2|\}$.

LEMMA 4. *In floating point number arithmetic with machine precision* ε_M

$$\mathrm{fl}(\sqrt{1+x^2}) = \sqrt{1+x^2}(1+2\varepsilon) + O(\varepsilon_M^2),$$

where $|\varepsilon| \leq \varepsilon_M$.

In the rest of the paper we denote the computed version of x by $\tilde{x}$. The next four bounds also follow from [1], but we provide the proof of the first bound.

LEMMA 5. *If we use the above angle formulas, then*

$$\begin{aligned}
\tilde{c}_l &= c_l(1+16\varepsilon_1) + O(\varepsilon_M^2),\\
\tilde{c}_r &= c_r(1+16\varepsilon_2) + O(\varepsilon_M^2),\\
\tilde{s}_l &= s_l(1+30\varepsilon_3) + O(\varepsilon_M^2),\\
\tilde{s}_r &= s_r(1+30\varepsilon_4) + O(\varepsilon_M^2),
\end{aligned}$$

where $|\varepsilon_i| \leq \varepsilon_M$ *for* $i = 1, 2, 3, 4$ *and* ε_M *is the machine precision.*

Proof. We go for the case where $|a| < |d|$ because this case gives a bigger bound for $\tilde{c}_l$. In the following, all $|\xi_i| \leq \varepsilon_M$. First, using our model of floating point number arithmetic,

$$\begin{aligned}
\tilde{r} &= \mathrm{fl}\left(\frac{(a+d)(a-d)}{b}\right)\\
&= \frac{(a+d)(a-d)}{b}(1+4\xi_1) + O(\varepsilon_M^2)\\
&= r(1+4\xi_1) + O(\varepsilon_M^2). \qquad (4)
\end{aligned}$$

Using the model of floating point number arithmetic, (4), and the fact that r and $-b$ have the same sign (and hence using Lemma 3), we have

$$\begin{aligned}
\tilde{\zeta}_r &= \mathrm{fl}\left(\frac{\tilde{r}-b}{2a}\right)\\
&= \frac{\tilde{r}-b}{2a}(1+2\xi_2) + O(\varepsilon_M^2)\\
&= \frac{r(1+4\xi_1)-b}{2a}(1+2\xi_2) + O(\varepsilon_M^2)\\
&= \frac{r(1+6\xi_3)-b(1+2\xi_2)}{2a} + O(\varepsilon_M^2)\\
&= \frac{(r-b)(1+6\xi_4)}{2a} + O(\varepsilon_M^2)\\
&= \zeta_r(1+6\xi_4) + O(\varepsilon_M^2). \qquad (5)
\end{aligned}$$

Using the model of floating point number arithmetic, Lemmas 4 and 3, and (5),

$$\begin{aligned}
\tilde{t}_r &= \mathrm{fl}\left(\frac{1}{\tilde{\zeta}_r + \mathrm{sign}(\tilde{\zeta}_r)\sqrt{1+\tilde{\zeta}_r^2}}\right)\\
&= \frac{1}{\tilde{\zeta}_r + \mathrm{sign}(\tilde{\zeta}_r)\sqrt{1+\tilde{\zeta}_r^2}}(1+4\xi_5) + O(\varepsilon_M^2)\\
&= \frac{1}{\zeta_r + \mathrm{sign}(\zeta_r)\sqrt{1+\zeta_r^2}}(1+10\xi_6) + O(\varepsilon_M^2)\\
&= t_r(1+10\xi_6) + O(\varepsilon_M^2). \qquad (6)
\end{aligned}$$

Using the model of floating point number arithmetic, (6), and the fact that at_r and $-b$ have the same sign (and hence using Lemma 3),

$$\begin{aligned}\tilde{t}_l &= \mathrm{fl}\left(\frac{a\tilde{t}_r - b}{d}\right)\\ &= \frac{a\tilde{t}_r(1+3\xi_7) - b(1+2\xi_8)}{d} + O(\varepsilon_M^2)\\ &= \frac{at_r(1+13\xi_9) - b(1+2\xi_8)}{d} + O(\varepsilon_M^2)\\ &= \frac{(at_r - b)(1+13\xi_{10})}{d} + O(\varepsilon_M^2)\\ &= t_l(1+13\xi_{10}) + O(\varepsilon_M^2). \qquad (7)\end{aligned}$$

Finally, using the model of floating point number arithmetic, Lemma 4, and (7),

$$\begin{aligned}\tilde{c}_l &= \mathrm{fl}\left(\frac{1}{\sqrt{1+\tilde{t}_l^2}}\right)\\ &= \frac{1}{\sqrt{1+\tilde{t}_l^2}}(1+3\xi_{11}) + O(\varepsilon_M^2)\\ &= \frac{1}{\sqrt{1+t_l^2}}(1+16\xi_{12}) + O(\varepsilon_M^2)\\ &= c_l(1+16\xi_{12}) + O(\varepsilon_M^2). \qquad \square\end{aligned}$$

The following lemma is easily proven from Lemma 2.2 in [5].

LEMMA 6. *Let*

$$A = \begin{bmatrix} x & z \\ z & y \end{bmatrix}$$

be positive definite and $z \neq 0$. *Suppose*

$$H = \begin{bmatrix} c & s \\ s & -c \end{bmatrix}$$

diagonalizes A. *Then*

$$|c|\,|s| \max\left\{\frac{\sqrt{x}}{\sqrt{y}}, \frac{\sqrt{y}}{\sqrt{x}}\right\} \leq 1.$$

Since H_l diagonalizes the positive definite matrix

$$CC^T = \begin{bmatrix} a^2+b^2 & bd \\ bd & d^2 \end{bmatrix},$$

applying Lemma 6 to CC^T gives

$$|c_l|\,|s_l| \max\left\{\frac{\sqrt{a^2+b^2}}{|d|}, \frac{|d|}{\sqrt{a^2+b^2}}\right\} \leq 1. \qquad (8)$$

4. Parallel Jacobi method. Let R be an $n \times n$ upper triangular matrix. By a left (right) reflector $\hat{H}_l$ ($\hat{H}_r$) for the pair $(i, i+1)$ of the matrix R $(1 \le i \le n-1)$ we mean an $n \times n$ identity matrix except that $\hat{H}_l(i : i+1, i : i+1) = H_l$ ($\hat{H}_r(i : i+1, i : i+1) = H_r$). We call $R \to \hat{H}_l^T R \hat{H}_r$ a transformation using the pair $(i, i+1)$. Note that this transformation annihilates the $(i, i+1)$ entry of R and that $\hat{H}_l^T R \hat{H}_r$ is still upper triangular.

The parallel Jacobi method is based on the fact that we can annihilate about $n/2$ elements of an $n \times n$ matrix simultaneously. We use the mobile parallel Jacobi method (MPJM) [6, pp. 349–369]. In the MPJM we consider two pairings, called pairing A and pairing B. We group the indices $1, 2, \ldots, n$ into the pairs

$$\text{pairing A} \quad (1,2), (3,4), \ldots, (l-1, l),$$

where $l = n$ if n is even and $l = n-1$ if n is odd, and

$$\text{pairing B} \quad (2,3), (4,5), \ldots, (m-1, m),$$

where $m = n-1$ if n is even and $m = n$ if n is odd. We will refer to a set of independent transformations performed simultaneously based on pairing A or pairing B as a batch. The MPJM is described in two sentences:

1. Perform batches of transformations using pairings A and B alternately.
2. After each transformation, interchange the participating rows and columns.

It takes n batches to annihilate all the off-diagonal entries once, that is, to perform one sweep. Note that the upper triangular structure of R is preserved throughout the process.

5. Error analysis. In this section we prove a bound on the roundoff error involved in any batch of the MPJM. We use this bound to show the method's accuracy in computing singular values and singular subspaces. We use $\approx$ ($\lesssim$) when an equality (inequality) holds up to the first order in machine precision or perturbation.

THEOREM 7. *Let R be an $n \times n$ invertible upper triangular matrix. We employ a batch of transformations according to MPJM as described in section* 4. *Let J_l (J_r) be the product of all the left (right) reflectors used in the batch, and $\tilde{R}_+$ be the computed version of $R_+ = J_l^T R J_r$ using the formulas in section* 3.

Then

$$\tilde{R}_+ = R_+ + \triangle R_+$$

with

$$\left\| R_+^{-1} (\triangle R_+) \right\|_2 \lesssim \begin{cases} 34 \varepsilon_M & \text{if} \quad n = 2, \\ 26\ n \left[\frac{1+1.6\omega}{\sigma_{\min}(D^{-1}R)} + \frac{1.5}{\sigma_{\min}(\tilde{D}_+^{-1}\tilde{R}_+)} \right] \varepsilon_M & \text{if} \quad n \ge 3. \end{cases}$$

Here D ($\tilde{D}_+$) is a diagonal matrix whose ith diagonal entry is the 2-norm of the ith row of R ($\tilde{R}_+$), and

$$\omega \equiv \max_i \omega_{i,i+1},$$

where

$$\omega_{i,i+1} \equiv \max \left(\max_{i+2 \le j \le n} \frac{|R(i,j)|}{|R(i,i)|}, \max_{i+2 \le j \le n} \frac{|R(i+1,j)|}{|R(i+1,i+1)|} \right)$$

and $(i, i+1)$ *belongs to pairing A or pairing B, depending on the current batch.*

Remark 3. If R is obtained from QR decomposition with column pivoting (for example, see [3, pp. 248–250]), then ω is bounded by 1 for R.

Proof. If $n = 2$, the matrix $\triangle R_+$ accounts for errors arising from our update formulas (3) for R. Since both R_+ and $\tilde{R}_+$ are diagonal matrices, $\triangle R_+$ is a diagonal matrix. Note that using Lemma 5 the diagonal elements $\tilde{a}_+$ and $\tilde{d}_+$ of $\tilde{R}_+$ are

$$\tilde{a}_+ = \text{fl}\left(\frac{\tilde{c}_l}{\tilde{c}_r}a\right) \approx \frac{c_l}{c_r}a(1+34\xi_1) = a_+(1+34\xi_1)$$

and

$$\tilde{d}_+ = \text{fl}\left(\frac{\tilde{c}_r}{\tilde{c}_l}d\right) \approx \frac{c_r}{c_l}d(1+34\xi_2) = d_+(1+34\xi_2),$$

where $|\xi_1|, |\xi_2| \le \varepsilon_M$. Since the diagonal entries of R_+ are a_+ and d_+, respectively, it follows that

$$\left\|R_+^{-1}(\triangle R_+)\right\|_2 \lesssim 34\varepsilon_M.$$

If $n \ge 3$, we write $R_1 = J_l^T(R + \triangle R_1)$ and $R_2 = [R_1 + \triangle R_2]J_r$. Here $\triangle R_1$ ($\triangle R_2$) represents the backward error from left (right) transformations, excepting the error in the diagonal blocks. Hence these two matrices are strictly block upper triangular. Since we update diagonal blocks by the formulas given in (3), we need another perturbation term $\triangle R_3$. Then we have

$$\begin{aligned}\tilde{R}_+ &= [J_l^T(R+\triangle R_1)+\triangle R_2]J_r + \triangle R_3\\ &= J_l^T R J_r + J_l^T(\triangle R_1)J_r + (\triangle R_2)J_r + \triangle R_3\\ &\equiv R_+ + \triangle R_+,\end{aligned}$$

where $\triangle R_+ = J_l^T(\triangle R_1)J_r + (\triangle R_2)J_r + \triangle R_3$. We bound

$$\eta = \left\|R_+^{-1}(\triangle R_+)\right\|_2$$

by $\eta_1 + \eta_2 + \eta_3$, where $\eta_1 = \|R_+^{-1}J_l^T(\triangle R_1)J_r\|_2$, $\eta_2 = \|R_+^{-1}(\triangle R_2)J_r\|_2$, and $\eta_3 = \|R_+^{-1}(\triangle R_3)\|_2$.

Note that

$$\begin{aligned}\eta_1 &= \left\|R_+^{-1}J_l^T(\triangle R_1)J_r\right\|_2\\ &= \left\|J_r^T R^{-1} J_l J_l^T(\triangle R_1)J_r\right\|_2\\ &= \left\|R^{-1}(\triangle R_1)\right\|_2\\ &\le \frac{\left\|D^{-1}(\triangle R_1)\right\|_2}{\sigma_{\min}(D^{-1}R)}.\end{aligned}$$

Now we bound $\|D^{-1}(\triangle R_1)\|_2$. Each individual left reflector affects two rows of R only. Suppose a particular reflector affects the rows i and $i+1$. Then we need only to consider elements in positions $(i : i+1, i+2 : n)$, because the diagonal block is updated separately. In particular, denote the elements of R in the positions $(i : i+1, j)$ by $\begin{bmatrix} g \\ h \end{bmatrix}$, where $i+2 \le j \le n$. Using Lemma 5, note that

$$\text{fl}\left(\tilde{H}_l^T \begin{bmatrix} g \\ h \end{bmatrix}\right) \approx H_l^T \begin{bmatrix} g \\ h \end{bmatrix} + \begin{bmatrix} 18\xi_3 g c_l + 32\xi_4 h s_l \\ 32\xi_5 g s_l + 18\xi_6 h c_l \end{bmatrix},$$

where $|\xi_3|, |\xi_4|, |\xi_5|, |\xi_6| \leq \varepsilon_M$. For backward error we set

$$\mathrm{fl}\left(\tilde{H}_l^T \begin{bmatrix} g \\ h \end{bmatrix}\right) = H_l^T \begin{bmatrix} g + \triangle g \\ h + \triangle h \end{bmatrix}.$$

Then

$$\begin{bmatrix} \triangle g \\ \triangle h \end{bmatrix} \approx \begin{bmatrix} c_l & s_l \\ s_l & -c_l \end{bmatrix} \begin{bmatrix} 18\xi_3 g c_l + 32\xi_4 h s_l \\ 32\xi_5 g s_l + 18\xi_6 h c_l \end{bmatrix}.$$

So

$$|\triangle g| \approx \left|18\xi_3 g c_l^2 + 32\xi_4 h c_l s_l + 32\xi_5 g s_l^2 + 18\xi_6 h c_l s_l\right|.$$

Hence it follows that

$$|\triangle g| \lesssim 32\varepsilon_M \left(|g| + \frac{25}{16}|c_l|\,|s_l|\,|h|\right).$$

Thus, writing

$$R(i:i+1, i:i+1) = \begin{bmatrix} a & b \\ 0 & d \end{bmatrix}$$

and using (8), the row-norm-scaled backward error for g satisfies

$$\begin{aligned} \frac{|\triangle g|}{\|R(i,:)\|_2} &\lesssim 32\varepsilon_M \left(\frac{|g|}{\|R(i,:)\|_2} + \frac{25}{16}|c_l|\,|s_l|\,\frac{|h|}{\|R(i,:)\|_2}\right) \\ &\leq 32\varepsilon_M \left(1 + \frac{25}{16}|c_l|\,|s_l|\,\frac{|d|}{\sqrt{a^2+b^2}}\frac{|h|}{|d|}\right) \\ &\leq 32\varepsilon_M \left(1 + \frac{25}{16}\frac{|h|}{|d|}\right). \end{aligned}$$

Similarly,

$$|\triangle h| \lesssim 32\varepsilon_M \left(|h| + \frac{25}{16}|c_l|\,|s_l|\,|g|\right),$$

and again using (8), the row-norm-scaled backward error for h is bounded by

$$\begin{aligned} \frac{|\triangle h|}{\|R(i+1,:)\|_2} &\lesssim 32\ \varepsilon_M \left(\frac{|h|}{\|R(i+1,:)\|_2} + \frac{25}{16}|c_l|\,|s_l|\,\frac{|g|}{\|R(i+1,:)\|_2}\right) \\ &\leq 32\ \varepsilon_M \left(1 + \frac{25}{16}|c_l|\,|s_l|\,\frac{\sqrt{a^2+b^2}}{|d|}\frac{|g|}{\sqrt{a^2+b^2}}\right) \\ &\leq 32\ \varepsilon_M \left(1 + \frac{25}{16}\frac{|g|}{\sqrt{a^2+b^2}}\right) \\ &\leq 32\ \varepsilon_M \left(1 + \frac{25}{16}\frac{|g|}{|a|}\right). \end{aligned}$$

By the definition of $\omega_{i,i+1}$, it follows that after the corresponding row vector-norm-scaling every element of $\triangle R_1$ in the positions $(i:i+1, i+2:n)$ is bounded to the

first order of ε_M by $32\varepsilon_M(1+\frac{25}{16}\omega_{i,i+1})$. Now it follows that

$$\begin{aligned}\left\|D^{-1}(\triangle R_1)\right\|_2 &\le \left\|D^{-1}(\triangle R_1)\right\|_F \\ &\lesssim 16\sqrt{2}\ n\ \varepsilon_M\left(1+\frac{25}{16}\omega\right) \\ &\le 23\ n\ \varepsilon_M(1+1.6\omega),\end{aligned}$$

and hence

$$\eta_1 \lesssim 23\ n\ \frac{1+1.6\omega}{\sigma_{\min}(D^{-1}R)}\ \varepsilon_M. \tag{9}$$

Since $R_+ = \tilde{R}_+ - \triangle R_+ = \tilde{R}_+(I - \tilde{R}_+^{-1}(\triangle R_+))$,

$$R_+^{-1} = (I - \tilde{R}_+^{-1}(\triangle R_+))^{-1}\tilde{R}_+^{-1} = \left[\sum_{l=0}^{\infty}(\tilde{R}_+^{-1}(\triangle R_+))^l\right]\tilde{R}_+^{-1}.$$

Hence

$$R_+^{-1}(\triangle R_2) = \left[\sum_{l=0}^{\infty}(\tilde{R}_+^{-1}(\triangle R_+))^l\right]\tilde{R}_+^{-1}(\triangle R_2) \approx \tilde{R}_+^{-1}(\triangle R_2).$$

Thus

$$\eta_2 = \left\|R_+^{-1}(\triangle R_2)J_r\right\|_2 = \left\|R_+^{-1}(\triangle R_2)\right\|_2 \approx \left\|\tilde{R}_+^{-1}(\triangle R_2)\right\|_2 \le \frac{\left\|\tilde{D}_+^{-1}(\triangle R_2)\right\|_2}{\sigma_{\min}(\tilde{D}_+^{-1}\tilde{R}_+)}.$$

Now we bound $\|\tilde{D}_+^{-1}(\triangle R_2)\|_2$. Each individual right reflector affects two columns of R_1 only. Suppose a particular reflector affects the columns j and $j+1$. Then we need only to consider elements in positions $(1:j-1, j:j+1)$, because the diagonal block is updated separately. In particular, denote the elements of R_1 in the positions $(i, j:j+1)$ by $[g\ h]$, where $1 \le i \le j-1$. Note that using Lemma 5

$$\mathrm{fl}\left(\begin{bmatrix} g & h \end{bmatrix}\tilde{H}_r\right) \approx \begin{bmatrix} g & h \end{bmatrix}H_r + \begin{bmatrix} 18\xi_7 g c_r + 32\xi_8 h s_r & 32\xi_9 g s_r + 18\xi_{10} h c_r \end{bmatrix},$$

where $|\xi_7|, |\xi_8|, |\xi_9|, |\xi_{10}| \le \varepsilon_M$. For backward error we set

$$\mathrm{fl}\left(\begin{bmatrix} g & h \end{bmatrix}\tilde{H}_r\right) = \begin{bmatrix} g+\triangle g & h+\triangle h \end{bmatrix}H_r.$$

Then

$$\begin{bmatrix} \triangle g & \triangle h \end{bmatrix} \approx \begin{bmatrix} 18\xi_7 g c_r + 32\xi_8 h s_r & 32\xi_9 g s_r + 18\xi_{10} h c_r \end{bmatrix}\begin{bmatrix} c_r & s_r \\ s_r & -c_r \end{bmatrix}.$$

Using the Cauchy–Schwarz inequality twice, we obtain

$$\begin{aligned}|\triangle g| &\approx \left|18\xi_7 g c_r^2 + 32\xi_8 h c_r s_r + 32\xi_9 g s_r^2 + 18\xi_{10} h c_r s_r\right| \\ &\le \sqrt{18^2+32^2}\ \varepsilon_M\ \sqrt{g^2+h^2}.\end{aligned}$$

Similarly, $|\triangle h| \lesssim \sqrt{18^2+32^2}\ \varepsilon_M\ \sqrt{g^2+h^2}$. Note that $\|R_1(i,:)\|_2 \ge \sqrt{g^2+h^2}$. Since right multiplication by an orthogonal matrix does not change row norms, $\|R_2(i,:)\|_2 \gtrsim$

$\sqrt{g^2+h^2}$. Since R_2 and $\tilde{R}_+$ differ only in diagonal blocks and g and h are not in a diagonal block, we also have $\|\tilde{R}_+(i,:)\|_2 \gtrsim \sqrt{g^2+h^2}$. So to the first order in ε_M, every element of $\tilde{D}_+^{-1}(\triangle R_2)$ is bounded by $\sqrt{18^2+32^2}\ \varepsilon_M$. Now it follows that

$$\left\|\tilde{D}_+^{-1}(\triangle R_2)\right\|_2 \le \left\|\tilde{D}_+^{-1}(\triangle R_2)\right\|_F \lesssim \sqrt{\frac{18^2+32^2}{2}}\ n\ \varepsilon_M \le 26\ n\ \varepsilon_M,$$

and hence

$$\eta_2 \lesssim \frac{26\ n}{\sigma_{\min}(\tilde{D}_+^{-1}\tilde{R}_+)}\ \varepsilon_M. \tag{10}$$

Finally, as before,

$$\eta_3 = \left\|R_+^{-1}(\triangle R_3)\right\|_2 \approx \left\|\tilde{R}_+^{-1}(\triangle R_3)\right\|_2 \le \frac{\left\|\tilde{D}_+^{-1}(\triangle R_3)\right\|_2}{\sigma_{\min}(\tilde{D}_+^{-1}\tilde{R}_+)}.$$

Now we bound $\|\tilde{D}_+^{-1}(\triangle R_3)\|_2$. The matrix $\triangle R_3$ accounts for errors arising from our update formulas (3) for the diagonal blocks. Actually there will only be errors in diagonal elements; that is, $\triangle R_3$ is a diagonal matrix. Note that using Lemma 5 the diagonal elements $\tilde{a}_+$ and $\tilde{d}_+$ of a typical diagonal block $(i:i+1, i:i+1)$ of $\tilde{R}_+$ are

$$\tilde{a}_+ = \mathrm{fl}\left(\frac{\tilde{c}_l}{\tilde{c}_r}a\right) \approx \frac{c_l}{c_r}a(1+34\xi_{11})$$

and

$$\tilde{d}_+ = \mathrm{fl}\left(\frac{\tilde{c}_r}{\tilde{c}_l}d\right) \approx \frac{c_r}{c_l}d(1+34\xi_{12}),$$

where $|\xi_{11}|, |\xi_{12}| \le \varepsilon_M$. The (i,i) and $(i+1,i+1)$ entries of $\triangle R_3$ are $34\xi_{11}a(c_l/c_r)$ and $34\xi_{12}d(c_r/c_l)$ to the first order in ε_M, respectively, and hence

$$\left\|\tilde{D}_+^{-1}(\triangle R_3)\right\|_2 \lesssim 34\varepsilon_M.$$

So

$$\eta_3 \lesssim \frac{34}{\sigma_{\min}(\tilde{D}_+^{-1}\tilde{R}_+)}\ \varepsilon_M. \tag{11}$$

The result of the theorem follows from (9), (10), and (11), using the fact that, if $n \ge 3$, we have $26n + 34 \le (1.5)(26n)$. ∎

Since the case when $n = 2$ is trivial, we will assume that $n \ge 3$ in what follows. Let $R_+ = U\Sigma V^T$ and $\tilde{R}_+ = R_+ + \triangle R_+ = \tilde{U}\tilde{\Sigma}\tilde{V}^T$ be the SVDs of R_+ and $\tilde{R}_+$, respectively. From Theorems 1 and 7 we obtain the following singular value relative perturbation bound:

$$\frac{|\tilde{\sigma}_i - \sigma_i|}{\sigma_i} \lesssim 26\ n\ \left[\frac{1+1.6\omega}{\sigma_{\min}(D^{-1}R)} + \frac{1.5}{\sigma_{\min}(\tilde{D}_+^{-1}\tilde{R}_+)}\right]\ \varepsilon_M. \tag{12}$$

We can also easily find a perturbation bound for the left singular subspace. From (1) and Theorem 7

$$\mathrm{s}(U_a, \tilde{U}_a) \lesssim 39n^2\left[\frac{1+1.6\omega}{\sigma_{\min}(D^{-1}R)} + \frac{1.5}{\sigma_{\min}(\tilde{D}_+^{-1}\tilde{R}_+)}\right]\frac{1}{\mathrm{sep}(\Sigma_a, \Sigma_b)}\ \varepsilon_M. \tag{13}$$

To get an upper bound on $\mathrm{s}(V_a, \tilde{V}_a)$ we need more work. Write $\tilde{U} = U + \triangle U, \tilde{\Sigma} = \Sigma + \triangle\Sigma$ and $\tilde{V} = V + \triangle V$. Then

$$R_+(\triangle V) + (\triangle R_+)V \approx U(\triangle\Sigma) + (\triangle U)\Sigma$$

by postmultiplying the SVD of $R_+ + \triangle R_+$ by $V + \triangle V$ and using the SVD of R_+. Premultiplication by $R_+^{-1} = V\Sigma^{-1}U^T$ gives us

$$\triangle V \approx V\Sigma^{-1}U^T(\triangle U)\Sigma + V\Sigma^{-1}(\triangle\Sigma) - R_+^{-1}(\triangle R_+)V.$$

Since $\mathrm{s}(V_a, \tilde{V}_a) = \|V_a^T\tilde{V}_b\|_2$, we find the expression of $V_a^T\tilde{V}_b$. Note that

$$V_a^T\tilde{V}_b = V_a^T(V_b + \triangle V_b) = V_a^T(\triangle V_b) = V_a^T(\triangle V)\begin{bmatrix} O \\ I_{n-k} \end{bmatrix},$$

so

$$V_a^T\tilde{V}_b \approx V_a^T\left[\; V\Sigma^{-1}U^T(\triangle U)\Sigma + V\Sigma^{-1}(\triangle\Sigma) - R_+^{-1}(\triangle R_+)V \;\right]\begin{bmatrix} O \\ I_{n-k} \end{bmatrix}.$$

Since $V_a^TV = \begin{bmatrix} I_k & O \end{bmatrix}$, by writing

$$U^T(\triangle U) = \begin{bmatrix} U_a^T \\ U_b^T \end{bmatrix}\begin{bmatrix} \triangle U_a & \triangle U_b \end{bmatrix} = \begin{bmatrix} U_a^T(\triangle U_a) & U_a^T(\triangle U_b) \\ U_b^T(\triangle U_a) & U_b^T(\triangle U_b) \end{bmatrix},$$

we obtain

$$V_a^T\tilde{V}_b \approx \Sigma_a^{-1}U_a^T(\triangle U_b)\Sigma_b + Y,$$

where

$$Y = \begin{bmatrix} I_k & O \end{bmatrix}\Sigma^{-1}(\triangle\Sigma)\begin{bmatrix} O \\ I_{n-k} \end{bmatrix} - V_a^TR_+^{-1}(\triangle R_+)V_b = -V_a^TR_+^{-1}(\triangle R_+)V_b.$$

Define $T = V^T\tilde{V} = V^T(V + \triangle V)$, whose (1,2) block is $V_a^T\tilde{V}_b$. Then for $1 \le i \le k$ and $k+1 \le j \le n$,

$$|t_{ij}| \lesssim \frac{\sigma_j}{\sigma_i}\;\left|s_{ij}\right| + \left|y_{i,j-k}\right|,$$

where s_{ij} is the (i,j) entry of $S = U^T\tilde{U}$, whose (1,2) block is $U_a^T\tilde{U}_b = U_a^T(\triangle U_b)$. From Theorem 2, for $1 \le i \le k$, $k+1 \le j \le n$,

$$|s_{ij}| \lesssim 3\;\left\|R_+^{-1}(\triangle R_+)\right\|_2\;\frac{\sigma_i\sigma_j}{\left|\sigma_i^2 - \sigma_j^2\right|},$$

and note that

$$\|Y\|_2 = \left\|-V_a^TR_+^{-1}(\triangle R_+)V_b\right\|_2 \le \left\|R_+^{-1}(\triangle R_+)\right\|_2.$$

So for $1 \le i \le k$ and $k+1 \le j \le n$,

$$\begin{aligned} |t_{ij}| &\lesssim 3\;\left\|R_+^{-1}(\triangle R_+)\right\|_2\;\frac{\sigma_j}{\sigma_i}\frac{\sigma_i\sigma_j}{\left|\sigma_i^2 - \sigma_j^2\right|} + \left\|R_+^{-1}(\triangle R_+)\right\|_2 \\ &= \left\|R_+^{-1}(\triangle R_+)\right\|_2\left(3\;\frac{\sigma_j^2}{\left|\sigma_i^2 - \sigma_j^2\right|} + 1\right). \end{aligned}$$

Since V_a corresponds to the k largest singular values,

$$3\ \frac{\sigma_j^2}{\left|\sigma_i^2-\sigma_j^2\right|}+1\le 3\ \frac{\sigma_i\sigma_j}{\left|\sigma_i^2-\sigma_j^2\right|}+1\le\frac{3}{\operatorname{sep}(\Sigma_a,\Sigma_b)}+1,$$

and hence

$$|t_{ij}|\lesssim\left\|R_+^{-1}(\triangle R_+)\right\|_2\ \left[\frac{3}{\operatorname{sep}(\Sigma_a,\Sigma_b)}+1\right]. \tag{14}$$

To provide an estimate for $\mathrm{s}(V_a,\tilde{V}_a)$ even when V_a corresponds to any simple, multiple, or clustered σ_i (see Remark 2), we consider the two cases, $\sigma_j\le\alpha\sigma_i$ and $\sigma_j>\alpha\sigma_i$, with

$$\alpha=\frac{\operatorname{sep}(\Sigma_a,\Sigma_b)+\sqrt{\operatorname{sep}^2(\Sigma_a,\Sigma_b)+4}}{2}.$$

Then in both cases one can verify that

$$3\ \frac{\sigma_j^2}{\left|\sigma_i^2-\sigma_j^2\right|}+1\le\frac{3}{\operatorname{sep}(\Sigma_a,\Sigma_b)}+4$$

for $1\le i\le k$ and $k+1\le j\le n$, and hence

$$|t_{ij}|\lesssim\left\|R_+^{-1}(\triangle R_+)\right\|_2\ \left[\frac{3}{\operatorname{sep}(\Sigma_a,\Sigma_b)}+4\right]. \tag{15}$$

Note that there is no essential difference between (14) and (15). Since the bound in (15) is bigger than that in (14), using (15) and Theorem 7 it follows that

$$\mathrm{s}(V_a,\tilde{V}_a)\lesssim 52n^2\left[\frac{1+1.6\omega}{\sigma_{\min}(D^{-1}R)}+\frac{1.5}{\sigma_{\min}(\tilde{D}_+^{-1}\tilde{R}_+)}\right]\left[\frac{1}{\operatorname{sep}(\Sigma_a,\Sigma_b)}+1\right]\ \varepsilon_M. \tag{16}$$

Remark 4. Note that the estimates for $\mathrm{s}(U_a,\tilde{U}_a)$ in (13) and $\mathrm{s}(V_a,\tilde{V}_a)$ in (16) hold for the SVD of the form $R=(UP)(P^T\Sigma P)(VP)^T$, where P is any permutation. So, if P is chosen in such a way that U_a and V_a correspond to (simple, multiple, or clustered) σ_i, the results (13) and (16) give estimates of how much the left and right singular subspaces belonging to σ_i are perturbed within one parallel batch. In this case $\operatorname{sep}(\Sigma_a,\Sigma_b)$ becomes the relative gap for σ_i in the set of singular values.

6. Numerical results. In this section we consider two numerical examples. We used MATLAB for all our computations. For each example we obtained approximate singular values and singular vectors in two different ways—first by applying MATLAB built-in function "svd" (MSVD) and then by applying the MPJM. Based on Theorem 1 and (1), we stopped the MPJM when

$$\left\|D_s^{-1}(R_s-D_s)\right\|_2\le 10^{-14},$$

where R_s is the upper triangular matrix obtained from R after s sweeps and D_s is the diagonal part of R_s. The results are summarized in Tables 1 through 8. In Tables 1 and 5, κ_s denotes the maximum value of

$$\frac{1+1.6\omega}{\sigma_{\min}(D^{-1}R)}+\frac{1.5}{\sigma_{\min}(\tilde{D}_+^{-1}\tilde{R}_+)}$$

TABLE 1
General information for MPJM of Example 1.

Sweep	$\|D_s^{-1}(R_s - D_s)\|_2$	κ_s
$s = 0$	3.07×10^{0}	
$s = 1$	2.87×10^{0}	3.07×10^{1}
$s = 2$	3.36×10^{-2}	1.90×10^{1}
$s = 3$	1.14×10^{-5}	2.53×10^{0}
$s = 4$	5.52×10^{-16}	2.50×10^{0}

TABLE 2
Relative accuracy of singular values for Example 1.

Singular values	Rel. accuracy of MSVD	Rel. accuracy of MPJM
1.0000×10^{0}	0	2.22×10^{-16}
1.0000×10^{0}	2.22×10^{-16}	2.22×10^{-16}
1.0000×10^{0}	0	6.66×10^{-16}
1.0000×10^{0}	1.11×10^{-16}	6.66×10^{-16}
1.0000×10^{-4}	4.54×10^{-14}	1.36×10^{-16}
1.0000×10^{-4}	2.97×10^{-13}	5.42×10^{-16}
1.0000×10^{-4}	2.01×10^{-13}	2.71×10^{-16}
1.0000×10^{-4}	4.79×10^{-13}	4.07×10^{-18}
1.0000×10^{-8}	3.71×10^{-9}	1.65×10^{-16}
1.0000×10^{-8}	2.67×10^{-10}	1.65×10^{-16}
1.0000×10^{-8}	2.03×10^{-9}	0
1.0000×10^{-8}	4.07×10^{-9}	3.31×10^{-16}
1.0000×10^{-12}	3.61×10^{-7}	2.02×10^{-16}
1.0000×10^{-12}	4.36×10^{-6}	0
1.0000×10^{-12}	3.54×10^{-6}	2.02×10^{-16}
9.9999×10^{-13}	8.06×10^{-6}	4.04×10^{-16}
1.3786×10^{-16}	2.38×10^{-6}	5.36×10^{-16}
1.0727×10^{-16}	1.11×10^{-5}	3.45×10^{-16}
8.0030×10^{-17}	4.52×10^{-6}	0
6.1551×10^{-17}	5.62×10^{-6}	0

TABLE 3
Accuracy of left singular subspaces for Example 1.

s$(U_a, \tilde{U}_a)$ from MSVD	s$(U_a, \tilde{U}_a)$ from MPJM
1.01×10^{-15}	1.09×10^{-19}
6.32×10^{-13}	1.45×10^{-19}
1.66×10^{-9}	2.09×10^{-19}
4.16×10^{-9}	1.55×10^{-19}
3.98×10^{-9}	1.77×10^{-19}

during the sth sweep. Note that κ_s measures to what extent errors within a sweep influence the accuracy of computed singular values and singular subspaces.

In order to check the accuracy of our computations (recall that the default precision of MATLAB is double precision), we computed the reference values in quadruple precision using Symbolic Math Toolbox (which is an extension of MATLAB).

Example 1. In this example a 20×20 nonsingular upper triangular matrix R is generated in the following way. First, we make the 20×20 diagonal matrix D whose diagonal entries are $d(1:4) = 1$, $d(5:8) = 10^{-4}$, $d(9:12) = 10^{-8}$, $d(13:16) = 10^{-12}$,

TABLE 4
Accuracy of right singular subspaces for Example 1.

$s(V_a, \tilde{V}_a)$ from MSVD	$s(V_a, \tilde{V}_a)$ from MPJM
4.26×10^{-16}	9.14×10^{-16}
1.03×10^{-12}	1.03×10^{-15}
8.36×10^{-9}	1.25×10^{-15}
3.50×10^{-5}	1.11×10^{-15}
3.50×10^{-5}	9.25×10^{-16}

TABLE 5
General information for MPJM of Example 2.

Sweep	$\|D_s^{-1}(R_s - D_s)\|_2$	κ_s
$s = 0$	1.60×10^{0}	
$s = 1$	1.01×10^{0}	1.52×10^{1}
$s = 2$	6.58×10^{-5}	5.57×10^{0}
$s = 3$	2.96×10^{-6}	2.50×10^{0}
$s = 4$	1.18×10^{-10}	2.50×10^{0}
$s = 5$	1.17×10^{-11}	2.50×10^{0}
$s = 6$	6.73×10^{-14}	2.50×10^{0}
$s = 7$	5.85×10^{-20}	2.50×10^{0}

TABLE 6
Relative accuracy of singular values for Example 2.

Singular values	Rel. accuracy of MSVD	Rel. accuracy of MPJM
1.0001×10^{-4}	2.71×10^{-16}	4.07×10^{-16}
1.0001×10^{-4}	1.36×10^{-16}	4.07×10^{-16}
1.0001×10^{-4}	2.71×10^{-16}	5.42×10^{-16}
1.0001×10^{-4}	4.07×10^{-16}	5.42×10^{-16}
1.0000×10^{-4}	4.07×10^{-16}	1.22×10^{-15}
1.0000×10^{-4}	1.36×10^{-16}	1.36×10^{-16}
1.0000×10^{-4}	1.36×10^{-16}	0
1.0000×10^{-4}	0	1.36×10^{-16}
9.9990×10^{-5}	1.36×10^{-16}	2.71×10^{-16}
9.9990×10^{-5}	0	0
9.9990×10^{-5}	2.71×10^{-16}	9.49×10^{-16}
9.9990×10^{-5}	1.08×10^{-15}	1.36×10^{-15}
1.0001×10^{-8}	2.65×10^{-13}	3.31×10^{-16}
1.0001×10^{-8}	2.25×10^{-13}	1.65×10^{-16}
1.0001×10^{-8}	2.70×10^{-13}	1.65×10^{-16}
1.0001×10^{-8}	1.02×10^{-13}	1.65×10^{-16}
1.0000×10^{-8}	4.96×10^{-13}	8.27×10^{-16}
1.0000×10^{-8}	4.28×10^{-13}	1.65×10^{-16}
1.0000×10^{-8}	7.99×10^{-13}	6.62×10^{-16}
1.0000×10^{-8}	2.81×10^{-13}	1.65×10^{-16}

and $d(17:20) = 10^{-16}$. Then we let $A = UDV^T$, where U and V are 20×20 random orthogonal matrices. Finally, R is obtained from A after QR decomposition with column pivoting.

Table 1 shows that κ_s remains on the order of 1 for each sweep. Thus we expect the computed singular values and singular subspaces are accurate up to machine capacity.

TABLE 7
Accuracy of left singular subspaces for Example 2.

$s(U_a, \tilde{U}_a)$ from MSVD	$s(U_a, \tilde{U}_a)$ from MPJM
3.88×10^{-12}	3.84×10^{-12}
4.27×10^{-12}	4.46×10^{-12}
3.69×10^{-12}	3.66×10^{-12}
1.56×10^{-8}	8.44×10^{-12}
1.56×10^{-8}	8.44×10^{-12}

TABLE 8
Accuracy of right singular subspaces for Example 2.

$s(V_a, \tilde{V}_a)$ from MSVD	$s(V_a, \tilde{V}_a)$ from MPJM
3.88×10^{-12}	3.84×10^{-12}
4.27×10^{-12}	4.46×10^{-12}
3.69×10^{-12}	3.67×10^{-12}
1.56×10^{-8}	8.44×10^{-12}
1.56×10^{-8}	8.44×10^{-12}

For singular values, the inequality (12) predicts relative accuracy on the order of machine precision. Indeed, Table 2 shows that the MPJM gives full relative accuracy for all singular values, while MSVD increasingly loses relative accuracy as singular values become smaller.

Notice that in this example there are five well-separated groups of multiple singular values, even though roundoff errors changes them into well-separated clusters. We check the accuracy of the singular subspace corresponding to each cluster of singular values.

For left singular subspaces, (13) predicts a bound of machine precision times an extra factor $1/\text{sep}(\Sigma_a, \Sigma_b)$, which is about 10^{-4} for each group. This is confirmed in Table 3, which displays the information according to decreasing size of associated singular values. We used permutation matrices to extract the needed singular subspace information (see Remark 4). Note that MSVD loses accuracy for subspaces associated with smaller singular values.

For right singular subspaces, (16) predicts a bound of machine precision times an extra factor $1/\text{sep}(\Sigma_a, \Sigma_b) + 1$, which is on the order of 1. This is confirmed in Table 4, while MSVD loses accuracy for subspaces associated with smaller singular values.

Example 2. In this example a 20×20 nonsingular upper triangular matrix R is generated as in Example 1, but starting with the 20×20 diagonal matrix D whose diagonal entries are $d(1:4) = 1.0001 \times 10^{-4}$, $d(5:8) = 10^{-4}$, $d(9:12) = 0.9999 \times 10^{-4}$, $d(13:16) = 1.0001 \times 10^{-8}$, and $d(17:20) = 10^{-8}$.

Table 5 shows that κ_s remains on the order of 1 for each sweep. Thus (12) predicts relative accuracy of singular values on the order of machine precision. Indeed, Table 6 shows that the MPJM gives full relative accuracy for all singular values, while MSVD loses relative accuracy for smaller singular values.

Notice that in this example there are five relatively poorly separated clusters of singular values. We check the accuracy of singular subspaces corresponding to each cluster of singular values, and we display the results as in Example 1.

For left singular subspaces, (13) predicts a bound of machine precision times an extra factor $1/\text{sep}(\Sigma_a, \Sigma_b)$, which is about 10^4 for this example. This is confirmed in

Table 7, while MSVD loses accuracy for subspaces associated with smaller singular values.

For right singular subspaces, (16) predicts a bound of machine precision times an extra factor $1/\text{sep}(\Sigma_a, \Sigma_b) + 1$, which is on the order of 10^4 in this example. This is confirmed in Table 8, while MSVD loses accuracy for subspaces associated with smaller singular values.

Acknowledgments. The authors acknowledge the anonymous referees' suggestions, which greatly improved the presentation of the paper.

REFERENCES

[1] A. Bojanczyk, L. Ewerbring, F. Luk, and P. van Dooren, *An accurate product SVD algorithm*, Signal Processing, 25 (1991), pp. 189–201.

[2] J. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.

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

[4] R. Mathias and K. Veselić, *A relative perturbation bound for positive definite matrices*, Linear Algebra Appl., 270 (1998), pp. 315–321.

[5] R. Mathias, *Accurate eigensystem computations by Jacobi methods*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 977–1003.

[6] D. Watkins, *Fundamentals of Matrix Computations*, John Wiley and Sons, New York, 1991.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 4, pp. 1001–1021

BREAKDOWN-FREE GMRES FOR SINGULAR SYSTEMS*

LOTHAR REICHEL† AND QIANG YE‡

Abstract. GMRES is a popular iterative method for the solution of large linear systems of equations with a square nonsingular matrix. When the matrix is singular, GMRES may break down before an acceptable approximate solution has been determined. This paper discusses properties of GMRES solutions at breakdown and presents a modification of GMRES to overcome the breakdown.

Key words. iterative method, Krylov subspace, singular matrix, linear system

AMS subject classifications. 65F10, 65F20

DOI. 10.1137/S0895479803437803

1. Introduction.

GMRES by Saad and Schultz [17] is one of the most popular methods for the iterative solution of large nonsymmetric linear systems of equations

$$Ax = b, \qquad A \in \mathbb{R}^{n\times n}, \qquad x, b \in \mathbb{R}^n. \tag{1.1}$$

The performance of the method is well understood when A is nonsingular, but the method also can be applied when A is singular. This paper focuses on the latter case. For notational simplicity, we choose the initial approximate solution of (1.1) to be $x_0 := 0$ and assume that the right-hand side vector b in (1.1) is normalized so that $\|b\| = 1$. Here and throughout this paper $\|\cdot\|$ denotes the Euclidean vector norm or the associated induced matrix norm.

The standard implementation of GMRES is based on the Arnoldi process. Application of k steps of the Arnoldi process to the matrix A with initial vector b yields the Arnoldi decomposition

$$AV_k = V_k H_k + f_k e_k^T, \tag{1.2}$$

where $H_k \in \mathbb{R}^{k\times k}$ is an upper Hessenberg matrix, $V_k \in \mathbb{R}^{n\times k}$, $V_k e_1 = b$, $V_k^T V_k = I_k$, $V_k^T f_k = 0$, I_k denotes the identity matrix of order k, and e_k is the kth axis vector. When $f_k \neq 0$, it is convenient to define the matrices

$$V_{k+1} := \left[V_k, \frac{f_k}{\|f_k\|}\right] \in \mathbb{R}^{n\times(k+1)}, \qquad \hat{H}_k := \begin{bmatrix} H_k \\ \|f_k\| e_k^T \end{bmatrix} \in \mathbb{R}^{(k+1)\times k} \tag{1.3}$$

and express (1.2) in the form

$$AV_k = V_{k+1}\hat{H}_k. \tag{1.4}$$

Note that $V_{k+1}^T V_{k+1} = I_{k+1}$. We assume that k is small enough so that at least one of the Arnoldi decompositions (1.2) or (1.4) exists. We will comment on the size of k below.

*Received by the editors November 16, 2003; accepted for publication (in revised form) by D. B. Szyld August 18, 2004; published electronically May 6, 2005.
http://www.siam.org/journals/simax/26-4/43780.html

†Department of Mathematical Sciences, Kent State University, Kent, OH 44242 (reichel@math.kent.edu). The work of this author was supported in part by National Science Foundation grant DMS-0107858.

‡Department of Mathematics, University of Kentucky, Lexington, KY 40506 (qye@ms.uky.edu). The work of this author was supported in part by National Science Foundation grant CCR-0098133.

It follows from (1.2) and the orthonormality of the columns of V_k that the latter form an orthonormal basis of the Krylov subspace

$$\mathcal{K}_k(A,b) := \text{span}\{b, Ab, \ldots, A^{k-1}b\}.$$

We will write $\mathcal{K}_k$ instead of $\mathcal{K}_k(A,b)$ when there is no ambiguity.

The kth iterate, x_k, determined by GMRES satisfies

$$\|b - Ax_k\| = \min_{z\in\mathcal{K}_k} \|b - Az\|, \qquad x_k \in \mathcal{K}_k.$$

Assuming that $f_k \neq 0$, the iterate x_k is computed by first solving the minimization problem in the right-hand side of

$$\min_{z\in\mathcal{K}_k} \|b - Az\| = \min_{y\in\mathbb{R}^k} \|b - AV_k y\| = \min_{y\in\mathbb{R}^k} \|e_1 - \hat{H}_k y\| \tag{1.5}$$

for $y_k \in \mathbb{R}^k$. Then x_k is given by

$$x_k := V_k y_k. \tag{1.6}$$

Since the subdiagonal entries of $\hat{H}_k$ are nonvanishing, the matrix $\hat{H}_k$ is of full rank, and therefore y_k is uniquely determined. We refer to Saad [16] and Saad and Schultz [17] for implementation details.

We say that the Arnoldi process (1.2) breaks down at step k if $f_k = 0$. Then the minimization problem (1.5) can be expressed as

$$\min_{z\in\mathcal{K}_k} \|b - Az\| = \min_{y\in\mathbb{R}^k} \|b - AV_k y\| = \min_{y\in\mathbb{R}^k} \|e_1 - H_k y\|, \tag{1.7}$$

and the solution y_k of the minimization problem in the right-hand side yields the solution $x := V_k y_k$ of (1.1). In this case, it is easy to show that if A is nonsingular, then H_k is nonsingular and y_k is uniquely determined.

When the matrix A is singular, the Arnoldi process may break down at step k with the upper Hessenberg matrix H_k in the decomposition (1.2) being singular. Let y_k denote the least-squares solution of minimal Euclidean norm of the minimization problem on the right-hand side of (1.7). The vector $x_k := V_k y_k$ is not guaranteed to solve the linear system of equations (1.1). The investigations [4, 5, 8] shed light on the properties of x_k, specifically on the question of whether x_k is a least-squares solution of (1.1). Related results also can be found in the review [14]. Several Krylov subspace methods for nonsymmetric singular systems are described in [1, 10, 11, 12, 18]. The present paper focuses on GMRES. Singular systems, several generalized inverses, and their applications are discussed in [2, 9].

We say that GMRES breaks down at step k if the Arnoldi process breaks down at step k. In this paper, we first discuss various properties of GMRES at breakdown, such as whether a solution is contained in the Krylov subspace and hence found by GMRES, and if not, what subspace contains a solution. Both consistent and inconsistent systems are considered. We then introduce a generalization of the Arnoldi decomposition that can be used when the (standard) Arnoldi process breaks down. We refer to GMRES based on the generalized Arnoldi decomposition as *breakdown-free GMRES* or simply *BFGMRES*. We also describe a breakdown-free variant of range restricted GMRES, which we refer to as *BFRRGMRES*. The (standard) RRGMRES method was introduced in [5]. Our interest in RRGMRES stems from the fact that

the method can determine more accurate approximations of the desired solution of large-scale discrete ill-posed problems than GMRES can; see [6] for illustrations. We remark that our approach to overcome breakdown of the Arnoldi process is related to but quite different from the technique described in [19] for avoiding breakdown of the nonsymmetric Lanczos process.

This paper is organized as follows. Section 2 discusses properties of approximate solutions determined by GMRES at breakdown when applied to the solution of consistent and inconsistent linear systems of equations with a singular matrix. Section 3 presents an algorithm for BFGMRES and discusses the minor modification required to obtain an algorithm for BFRRGMRES. Some properties of (BF)RRGMRES are also discussed. A few numerical examples are presented in section 4.

We remark that linear systems of equations with a numerically singular matrix arise, for instance, in the context of ill-posed problems (see [6, 7]) and when computing the steady state distribution of finite Markov chains; see, e.g., [15]. Furthermore, overdetermined systems of equations with n rows and m columns, where $n > m$, can be brought into the form (1.1) by appending $n - m$ zero columns to the matrix. The matrix A so obtained is singular, and the linear system of equations can be solved by BFGMRES or BFRRGMRES. A comparison of this approach with application of the conjugate gradient method to the normal equations is presented in section 4. Underdetermined linear systems of equations can also be solved by BFGMRES or BFRRGMRES by first appending an appropriate number of zero rows to the matrix.

2. Breakdown of GMRES. We first discuss breakdown of the (standard) Arnoldi process in some detail and introduce the notions of benign and hard breakdowns. There is a positive integer N, such that

$$\dim(\mathcal{K}_k) = \begin{cases} k, & 1 \le k \le N, \\ N, & k \ge N+1. \end{cases}$$

This easily can be seen by using the Jordan form of A. Clearly, $N \le n$.

For $k \le N$, the Arnoldi decomposition (1.2) exists. We distinguish two cases:

1. If $Av_k \notin \mathcal{K}_k$, then $f_k \neq 0$ in (1.2). It follows that the decomposition (1.4) exists, and the columns of the matrix V_{k+1} form an orthonormal basis of $\mathcal{K}_{k+1}$, the matrix $\hat{H}_k$ is of full rank, and the minimization problem in the right-hand side of (1.5) has a unique solution y_k, which by (1.6) determines x_k, the kth iterate generated by GMRES. It follows from $\dim(\mathcal{K}_{k+1}) = k+1$ that $Ax_k \neq b$.
2. If $Av_k \in \mathcal{K}_k$, then $f_k = 0$ in the Arnoldi decomposition (1.2). We have $\dim(\mathcal{K}_{k+1}) = k$, and therefore $k = N$. The Arnoldi process and GMRES break down. Again, we distinguish two cases:
   (a) If $\dim(A\mathcal{K}_N) = N$, then $\operatorname{rank}(H_N) = N$, and the Nth iterate, x_N, generated by GMRES is determined by first solving the minimization problem in the right-hand side of (1.7), with k replaced by N, for y_N, and then computing x_N from (1.6) with k replaced by N. Since

   $$\operatorname{span}\{b\} + A\mathcal{K}_N = \mathcal{K}_{N+1}, \qquad \dim(\mathcal{K}_{N+1}) = N,$$

   we have that $b \in A\mathcal{K}_N$. Thus, $Ax_N = b$. Therefore, this is referred to as a *benign breakdown*; the solution has been found when the breakdown occurs, just like when the matrix A is nonsingular.
   (b) If $\dim(A\mathcal{K}_N) < N$, then $\operatorname{rank}(H_N) < N$. Let y_N be the solution of minimal norm of the least-squares problem in the right-hand side of

(1.7), and determine the iterate x_N by (1.6) with k replaced by N. Note that $Ax_N \neq b$ because $b \notin A\mathcal{K}_N$. We refer to this as a *hard breakdown.*

We remark that our classification of breakdowns is slightly different from that of Brown and Walker [4].

Next, we characterize the approximate solutions of (1.1) that can be determined by GMRES when a hard breakdown occurs. Throughout this paper $\mathcal{N}(M)$ denotes the null space and $\mathcal{R}(M)$ denotes the range of the matrix M. We consider consistent and inconsistent systems separately.

2.1. Consistent systems. Ipsen and Meyer [14] showed that Krylov subspace iterative methods, such as GMRES, are able to determine a solution of the linear system of equations (1.1) if and only if $b \in \mathcal{R}(A^D)$, where A^D denotes the Drazin inverse of A; see (2.10) below for a definition. The following theorem complements this result; it discusses the form of the solution when the right-hand side b is a general vector in $\mathcal{R}(A)$.

THEOREM 2.1. *Let the matrix A be singular and assume that the linear system of equations* (1.1) *is consistent. Apply GMRES with initial approximate solution $x_0 := 0$ to the solution of* (1.1) *and assume that a hard breakdown occurs at step N. If $A^N b \neq 0$, then any solution x of* (1.1) *can be expressed as*

$$x = \hat{x} + u, \tag{2.1}$$

where $\hat{x} \in \mathcal{K}_{N-1}$ and $u \in \mathcal{N}(A^\ell)\backslash\{0\}$ for some integer ℓ with $2 \leq \ell \leq N$. If instead $A^N b = 0$, then any solution of (1.1) *belongs to $\mathcal{N}(A^{N+1})$.*

Proof. We first consider the case when $A^N b \neq 0$. Since $\dim(\mathcal{K}_N) = N$ and $\dim(A\mathcal{K}_N) < N$, the vector $A^N b$ is a linear combination of the vectors $\{A^j b\}_{j=1}^{N-1}$. Let ℓ be the largest integer with $2 \leq \ell \leq N$, such that

$$\alpha_{\ell-1}A^{\ell-1}b + \alpha_\ell A^\ell b + \cdots + \alpha_{N-1}A^{N-1}b + A^N b = 0 \tag{2.2}$$

for some coefficients $\alpha_{\ell-1}, \alpha_\ell, \ldots, \alpha_{N-1}$. Clearly, $\alpha_{\ell-1} \neq 0$.

Let x be a solution of (1.1). Then (2.2) yields

$$A^\ell x + \frac{\alpha_\ell}{\alpha_{\ell-1}}A^\ell b + \cdots + \frac{\alpha_{N-1}}{\alpha_{\ell-1}}A^{N-1}b + \frac{1}{\alpha_{\ell-1}}A^N b = 0$$

or, equivalently,

$$A^\ell \left(x + \frac{\alpha_\ell}{\alpha_{\ell-1}}b + \cdots + \frac{\alpha_{N-1}}{\alpha_{\ell-1}}A^{N-\ell-1}b + \frac{1}{\alpha_{\ell-1}}A^{N-\ell}b \right) = 0. \tag{2.3}$$

Let

$$\hat{x} := -\frac{\alpha_\ell}{\alpha_{\ell-1}}b - \cdots - \frac{\alpha_{N-1}}{\alpha_{\ell-1}}A^{N-\ell-1}b - \frac{1}{\alpha_{\ell-1}}A^{N-\ell}b, \qquad u := x - \hat{x}.$$

Clearly, $\hat{x} \in \mathcal{K}_{N-\ell+1} \subset \mathcal{K}_{N-1}$, and it follows from (2.3) that $u \in \mathcal{N}(A^\ell)$. Since $A\hat{x} \neq b$, we have $u \neq 0$.

We turn to the case when $A^N b = 0$. With $b = Ax$, we have $A^{N+1}x = 0$, which shows that $x \in \mathcal{N}(A^{N+1})$. ∎

The matrix A is said to have index p if its largest Jordan block associated with the eigenvalue zero is of order p. It follows that if A has index p, then $\mathcal{N}(A^j) = \mathcal{N}(A^p)$

for all integers $j \geq p$. Then by Theorem 2.1 any solution x belongs to a subspace extended from the Krylov subspace, i.e.,

$$x \in \mathcal{K}_{N-1} + \mathcal{N}(A^p). \tag{2.4}$$

We note that only the Krylov subspace $\mathcal{K}_{N-1}$ is needed. This is different from the situation of benign breakdown, where the solution of (1.1) belongs to $\mathcal{K}_N$. This fact will be used in our extension of GMRES described in section 3.

Brown and Walker [4, Theorem 2.6] show that if the linear system of equations (1.1) is consistent and

$$\mathcal{N}(A) \cap \mathcal{R}(A) = \{0\}, \tag{2.5}$$

then GMRES applied to (1.1) with initial approximate solution $x_0 := 0$ determines a solution. This result is a corollary to the theorem above. Note that condition (2.5) is equivalent to A having index one.

COROLLARY 2.2. *Let A be a singular matrix of index one, and assume that the linear system of equations* (1.1) *is consistent. Then hard breakdown cannot occur.*

Proof. We use the notation of Theorem 2.1 and its proof. Assume that a hard breakdown occurs at step N of GMRES. First consider the situation when $A^N b \neq 0$. Theorem 2.1 shows that $\hat{x} = x - u$ with $u \in \mathcal{N}(A^\ell) = \mathcal{N}(A)$. Therefore, $A\hat{x} = Ax - Au = b$, which is a contradiction.

We turn to the case when $A^N b = 0$ and $b \neq 0$. Then $x \in \mathcal{N}(A^{N+1}) = \mathcal{N}(A)$. Hence, $Ax = 0$, which is a contradiction. ∎

We consider an application of Corollary 2.2.

Example 2.1. Let $\tilde{A} \in \mathbb{R}^{n \times \ell}$, with $\ell < n$, and assume that the leading $\ell \times \ell$ principal submatrix of $\tilde{A}$ is nonsingular. Let $b \in \mathcal{R}(\tilde{A})$. We are interested in computing the solution of the consistent linear system of equations

$$\tilde{A}\tilde{x} = b. \tag{2.6}$$

Assume that a function for the evaluation of matrix-vector products with the matrix $\tilde{A}$ is available, but that the entries of the matrix are not explicitly known. It then may be attractive to solve (2.6) by an iterative method. The standard iterative method for this task is the conjugate gradient method applied to the normal equations associated with (2.6), using the CGLS or LSQR algorithms; see, e.g., [3]. These algorithms require the evaluation of matrix-vector products with both the matrices $\tilde{A}$ and $\tilde{A}^T$. If only a function for the evaluation of matrix-vector products with $\tilde{A}$ is available, but not with $\tilde{A}^T$, then we may consider using GMRES, which does not require $\tilde{A}^T$. We note that the cost per iteration for GMRES increases rapidly with the iteration and restarts are needed in practical implementations. The fact that the matrix $\tilde{A}$ is not square can be overcome by padding $\tilde{A}$ with $n - \ell$ trailing zero columns. This yields an $n \times n$ matrix, which we denote by A, and we obtain a linear system of equations of the form (1.1). GMRES then is applied to compute an approximate solution of this system. Note that zero is an eigenvalue of A of algebraic multiplicity $n - \ell$; this can be seen from the Schur form. Moreover, the axis vectors $e_{\ell+1}, e_{\ell+2}, \ldots, e_n$ are in $\mathcal{N}(A)$. It follows that A has index one, and by Corollary 2.2, GMRES cannot suffer from a hard breakdown.

Let $x_k \in \mathbb{R}^n$ denote the kth iterate determined by GMRES. The first ℓ entries of x_k yield an approximate solution of (2.6). The zero columns of A, of course, do not have to be stored.

We remark that the requirement that $\tilde{A}$ have a nonsingular $\ell \times \ell$ leading principal submatrix secures that the matrix is of full rank. Conversely, if $\tilde{A}$ is of full rank, then there is a row-permutation such that the leading principal $\ell \times \ell$ submatrix is nonsingular. We also observe that different row permutations could lead to a very different performance of GMRES, because the spectrum of A may change as the rows are interchanged. A comparison of the convergence behavior of CGLS and GMRES when applied to the solution of linear systems of equations of the form (2.6) is presented in section 4.

We also consider the special case when a breakdown occurs when the dimension of the Krylov subspace N is equal to the rank of A. This should be compared with Theorem 2.7 below where interestingly a much stronger result exists for inconsistent systems.

THEOREM 2.3. *Let the matrix $A \in \mathbb{R}^{n\times n}$ be of rank $N < n$ and assume that the linear system of equations* (1.1) *is consistent. Apply GMRES with initial approximate solution $x_0 := 0$ to the solution of* (1.1). *Assume that GMRES breaks down at step N. If $\dim(A\mathcal{K}_N) = N$, then GMRES determines a solution of* (1.1) *at breakdown. If, instead, $\dim(A\mathcal{K}_N) < N$, then* (1.1) *has a solution in $\mathcal{K}_N + \mathcal{R}(A^T)$.*

Proof. The Arnoldi process breaks down at step N and yields the decomposition

$$AV_N = V_N H_N, \qquad V_N e_1 = b. \tag{2.7}$$

If $\dim(A\mathcal{K}_N) = N$, then the breakdown is benign and GMRES determines a solution of (1.1).

We turn to the case when $\dim(A\mathcal{K}_N) < N$. Then the upper Hessenberg matrix H_N in the decomposition (2.7) is singular. Since H_N has positive subdiagonal entries, $\operatorname{rank}(H_N) = N - 1$. Let $u \in \mathcal{N}(H_N^T)$ be of unit length and introduce $v := A^\dagger V_N u$, where $A^\dagger$ denotes the Moore–Penrose pseudoinverse of A. Note that $v \in \mathcal{R}(A^T)$. Since $b \in \mathcal{R}(A)$, we have that $\mathcal{R}(V_N) \subset \mathcal{R}(A)$. Therefore, $Av = V_N u$ and it follows that $V_N^T A v = u$. We seek a solution of (1.1) of the form

$$x = V_N y + v\eta, \qquad y \in \mathbb{R}^N, \qquad \eta \in \mathbb{R}.$$

Substituting this expression into (1.1) yields the equation $AV_N y + Av\eta = b$, which, using (2.7), can be seen to be equivalent to

$$[H_N, u] \begin{bmatrix} y \\ \eta \end{bmatrix} = e_1. \tag{2.8}$$

Since the matrix $[H_N, u] \in \mathbb{R}^{N\times(N+1)}$ is of full rank, (2.8) has a solution $\{\hat{y}, \hat{\eta}\}$, which gives the solution $\hat{x} := V_N \hat{y} + v\hat{\eta}$ of (1.1). ∎

In the second case of the theorem, i.e., when $\dim(A\mathcal{K}_N) < N$, a solution of (1.1) can be determined by a modification of GMRES that minimizes the residual error over $\mathcal{K}_N + \mathcal{R}(A^T)$.

2.2. Inconsistent systems. First, we note that, for inconsistent systems, H_N in the Arnoldi decomposition determined at breakdown must be singular, because otherwise a solution to (1.1) would be obtained. Therefore, only hard breakdown can occur. We consider the computation of a least-squares solution of (1.1) and formulate our results in terms of the Drazin inverse of A. Let A have the representation

$$A = C \begin{bmatrix} J_0 & 0 \\ 0 & J_1 \end{bmatrix} C^{-1}, \tag{2.9}$$

where the matrix $C \in \mathbb{C}^{n\times n}$ is invertible, the matrix J_0 consists of all Jordan blocks associated with the eigenvalue zero, and the matrix J_1 consists of all Jordan blocks associated with nonvanishing eigenvalues. The Drazin inverse of A is given by

$$A^D := C \begin{bmatrix} 0 & 0 \\ 0 & J_1^{-1} \end{bmatrix} C^{-1}. \tag{2.10}$$

See [9, Chapter 7] for properties of this generalized inverse. We note that if A has index p, then

$$\mathcal{N}(A^p) = \mathcal{N}(A^D).$$

THEOREM 2.4. *Let the singular matrix A have index p. Assume that a hard breakdown occurs at step N when GMRES is applied to* (1.1) *with initial approximate solution $x_0 := 0$, and that $A^N b \neq 0$. Then any least-squares solution x of* (1.1) *can be written in the form*

$$x = \hat{x} + u - A^D r, \tag{2.11}$$

where $\hat{x} \in \mathcal{K}_{N-1}$, $u \in \mathcal{N}(A^D) = \mathcal{N}(A^p)$, and $r := b - Ax \in \mathcal{N}(A^T)$ is the residual vector associated with x.

Proof. Similarly as in the proof of Theorem 2.1, let ℓ be the largest integer with $2 \leq \ell \leq N$, such that (2.2) holds. It follows that

$$A^{\ell-1}\left(b + \frac{\alpha_\ell}{\alpha_{\ell-1}}Ab + \cdots + \frac{\alpha_{N-1}}{\alpha_{\ell-1}}A^{N-\ell}b + \frac{1}{\alpha_{\ell-1}}A^{N-\ell+1}b\right) = 0.$$

Since $\mathcal{N}(A^{\ell-1}) \subset \mathcal{N}(A^p)$, we also have

$$A^{p}\left(b + \frac{\alpha_\ell}{\alpha_{\ell-1}}Ab + \cdots + \frac{\alpha_{N-1}}{\alpha_{\ell-1}}A^{N-\ell}b + \frac{1}{\alpha_{\ell-1}}A^{N-\ell+1}b\right) = 0. \tag{2.12}$$

Let x be a least-squares solution of (1.1) and introduce the associated residual vector $r := b - Ax$. Substituting $b = r + Ax$ into (2.12) yields

$$A^{p}\left(Ax + \frac{\alpha_\ell}{\alpha_{\ell-1}}Ab + \cdots + \frac{\alpha_{N-1}}{\alpha_{\ell-1}}A^{N-\ell}b + \frac{1}{\alpha_{\ell-1}}A^{N-\ell+1}b\right) = -A^p r,$$

and, therefore,

$$A^{p+1}\left(x + \frac{\alpha_\ell}{\alpha_{\ell-1}}b + \cdots + \frac{\alpha_{N-1}}{\alpha_{\ell-1}}A^{N-\ell-1}b + \frac{1}{\alpha_{\ell-1}}A^{N-\ell}b\right) = -A^p r. \tag{2.13}$$

Let

$$\hat{x} := -\frac{\alpha_\ell}{\alpha_{\ell-1}}b - \cdots - \frac{\alpha_{N-1}}{\alpha_{\ell-1}}A^{N-\ell-1}b - \frac{1}{\alpha_{\ell-1}}A^{N-\ell}b, \qquad w := x - \hat{x}.$$

Then $\hat{x} \in \mathcal{K}_{N-\ell+1} \subset \mathcal{K}_{N-1}$ and

$$A^{p+1}w = -A^p r. \tag{2.14}$$

The linear system of equations (2.14) is consistent, and any solution can be expressed as $w = -A^D r + u$, where A^D denotes the Drazin inverse of A and $u \in \mathcal{N}(A^p)$. We remark that $\mathcal{R}(A^D) + \mathcal{N}(A^D)$ makes up all of the n-space. ∎

The following corollary considers the situation when, in addition to the conditions of Theorem 2.4,

$$\mathcal{N}(A^T) \subset \mathcal{N}(A^D). \tag{2.15}$$

In this case, the following result, which is similar to our result for the consistent case, holds.

COROLLARY 2.5. *Let the singular matrix A have index p and assume that a hard breakdown occurs at step N when GMRES is applied to* (1.1) *with initial approximate solution $x_0 := 0$. Let x be a least-squares solution of* (1.1) *and assume that* (2.15) *holds. If $A^N b \neq 0$, then x can be written in the form*

$$x = \hat{x} + u, \tag{2.16}$$

where $\hat{x} \in \mathcal{K}_{N-1}$ and $u \in \mathcal{N}(A^D) = \mathcal{N}(A^p)$. If, instead, $A^N b = 0$, then $x \in \mathcal{N}(A^D)$.

Proof. First assume that $A^N b \neq 0$. Let $r := b - Ax$ denote the residual vector associated with the least-squares solution x of (1.1). Then $r \in \mathcal{N}(A^T)$ and (2.15) yields $A^D r = 0$. Equation (2.16) now follows from (2.11).

If $A^N b = 0$, then $A^p b = 0$. Therefore,

$$0 = A^p b = A^p(r + Ax) = A^p r + A^{p+1}x = A^{p+1}x,$$

and $x \in \mathcal{N}(A^D)$ follows from $\mathcal{N}(A^{p+1}) = \mathcal{N}(A^D)$. ∎

Let

$$A = QSQ^* \tag{2.17}$$

be a Schur decomposition; i.e., $S \in \mathbb{C}^{n\times n}$ is upper triangular, $Q \in \mathbb{C}^{n\times n}$ is unitary, and the superscript $*$ denotes transposition and complex conjugation. Order the eigenvalues and partition

$$S = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix} \tag{2.18}$$

so that all diagonal entries of S_{11} are zero and the diagonal entries of S_{22} are nonvanishing. Using $\mathcal{N}(A^p) = \mathcal{N}(A^n)$ and $S_{11}^n = 0$, we can show that S_{11} J_1 in $S_{11}^p = 0$. $\mathcal{N}(A^T) \subset \mathcal{N}(A^p)$ is equivalent to $\mathcal{N}(S_{11}^T) \subset \mathcal{N}(S_{12}^T)$.

The following result by Brown and Walker [4, Theorem 2.4] can be shown in a similar manner as Theorem 2.4 above. We include a proof for completeness.

COROLLARY 2.6. *Let A be a singular matrix, such that $\mathcal{N}(A) = \mathcal{N}(A^T)$. Apply GMRES to* (1.1) *with initial approximate solution $x_0 := 0$. Then GMRES determines a least-squares solution at breakdown.*

Proof. Using the Schur decomposition (2.17) of A and the partitioning (2.18), it can be shown that the condition $\mathcal{N}(A^T) = \mathcal{N}(A)$ implies that $S_{11} = 0$ and $S_{12} = 0$. Hence, A has index $p = 1$. Thus, (2.15) holds, and therefore the conditions of Corollary 2.5 are satisfied. Assume that GMRES breaks down at step N. If $A^N b \neq 0$, then Corollary 2.5 shows that a least-squares solution x can be expressed as $x = \hat{x}+u$, where $\hat{x} \in \mathcal{K}_{N-1}$ and $u \in \mathcal{N}(A^p) = \mathcal{N}(A)$. It follows that $b - A\hat{x} = b - Ax$ and, therefore, $\hat{x}$ is a least-squares solution of (1.1). Since GMRES minimizes the Euclidean norm of the residual error over $\mathcal{K}_N$, GMRES will determine a least-squares solution.

If, instead, $A^N b = 0$, then $Ab = 0$, and therefore $A^T b = 0$. Hence, $x = 0$ is a least-squares solution and so is any multiple of b. GMRES breaks down at step one and yields the least-squares problem

$$\min_{y \in \mathbb{R}} |H_1 y - 1| \tag{2.19}$$

with $H_1 = 0$, where we have used that $\|b\| = 1$. The minimal-norm least-squares solution $y := 0$ gives the least-squares solution $x := 0$ of (1.1). A least-squares solution $y \neq 0$ of (2.19) yields the least-squares solution $x := yb$ of (1.1). ∎

Corollary 2.6 holds for any right-hand side vector b in (1.1) but requires that $\mathcal{N}(A) = \mathcal{N}(A^T)$. We remark that GMRES often can find a least-squares solution even when this condition does not hold (see Example 4.1 below). The following result, first stated in [5, Theorem 2.2], explains this observed behavior. It deals with the most typical situation of breakdown, i.e., a breakdown at step $\operatorname{rank}(A) + 1$, which is the upper bound of the dimension of the Krylov subspace as $\mathcal{K}_{k+1} = \operatorname{span}\{b\} + A\mathcal{K}_k$ and $b \notin \mathcal{R}(A)$.

THEOREM 2.7. *Let the matrix $A \in \mathbb{R}^{n \times n}$ be of rank $N < n$ and apply GMRES with initial approximate solution $x_0 := 0$ to the solution of* (1.1). *If GMRES breaks down at step $N + 1$, then GMRES determines a least-squares solution.*

Proof. An index is incorrect in the proof in [5]; the index is not consistent with the definition of breakdown in [5], which is different from the definition in the present paper. We therefore provide a proof here.

The Arnoldi process generates an orthonormal basis of the Krylov subspace $\mathcal{K}_{N+1} = \operatorname{span}\{b\} + A\mathcal{K}_N$ before breakdown. It follows that $\dim(A\mathcal{K}_N) = N$. Suppose that GMRES does not determine a least-squares solution of (1.1); i.e., there is no $x \in \mathcal{K}_{N+1}$ that satisfies the normal equations $A^T Ax = A^T b$. In other words,

$$A^T b \notin A^T A\mathcal{K}_{N+1}. \tag{2.20}$$

By Lemma 2.1 in [5], or by [4, pp. 40–41], $\dim(A^T A\mathcal{K}_{N+1}) = \dim(A\mathcal{K}_{N+1})$, and since $\dim(A\mathcal{K}_{N+1}) \geq \dim(A\mathcal{K}_N) = N$, we obtain that $\dim(A^T A\mathcal{K}_{N+1}) \geq N$. It now follows from (2.20) that $A^T\mathcal{K}_{N+2} = \operatorname{span}\{A^T b\} + A^T A\mathcal{K}_{N+1}$ is of dimension at least $N + 1$. However, $\operatorname{rank}(A^T) = \operatorname{rank}(A) = N$. Therefore, $\dim(A^T\mathcal{K}_{N+2}) \leq N$. This contradiction shows that $\mathcal{K}_{N+1}$ does contain a least-squares solution of the normal equation. Hence, GMRES determines a least-squares solution. ∎

The conditions of Theorem 2.7 hold if $p(A)b \neq 0$ for any polynomial p of degree less than or equal to N. This is the case, for example, when A has distinct nonzero eigenvalues, at most one zero eigenvalue with a nontrivial Jordan block, and each eigenvector and its associated Jordan chain have a nonzero component in b. We also note that the conditions can be satisfied only if the linear system (1.1) is inconsistent, because otherwise $\mathcal{R}(V_{N+1}) \subset \mathcal{R}(A)$. But this inclusion cannot hold since $\dim(\mathcal{R}(A)) = N$.

Example 2.2. Let $\tilde{A}$ be a matrix of the same kind as in Example 2.1, and assume that $b \in \mathbb{R}^n$ is not in $\mathcal{R}(A)$. We are interested in computing the solution of the least-squares problem

$$\min_{\tilde{x} \in \mathbb{R}^\ell} \|\tilde{A}\tilde{x} - b\|. \tag{2.21}$$

Similarly as in Example 2.1, we define the matrix $A \in \mathbb{R}^{n \times n}$ by padding $\tilde{A}$ with $n - \ell$ trailing zero columns. We obtain a linear system of equations of the form (1.1). If

GMRES applied to this system with initial approximate solution $x_0 := 0$ does not break down until step $\ell + 1$, then according to Theorem 2.7 a least-squares solution of (1.1) has been determined. The first ℓ components of the computed solution make up a least-squares solution of (1.1).

This example illustrates that it may be possible to determine a solution of (2.21) by (standard) GMRES. The breakdown-free GMRES method of the following section is useful when (standard) GMRES breaks down before step $\ell + 1$.

3. Breakdown-free GMRES. This section presents an extension of GMRES to overcome breakdown. We comment on a breakdown-free variant of RRGMRES at the end of the section.

From our discussions in section 2, when GMRES suffers a hard breakdown at step N, the Krylov subspace $\mathcal{K}_N$ does not contain a solution of the linear system (1.1); however, as Theorem 2.1 shows, any solution belongs to $\mathcal{K}_{N-1}+\mathcal{N}(A^p)$. This suggests that, to compute a solution, the Krylov subspace $\mathcal{K}_{N-1}$ has to be extended to capture the component of the solution in $\mathcal{N}(A^p)$, which is an eigenvector of A^p corresponding to the eigenvalue zero. This eigenvector can be approximated from a Krylov subspace generated by a new vector. Therefore, at every breakdown of the Arnoldi process, we generate a new Krylov subspace and add it to the available one(s). Then we seek an approximation from the extended subspace. An implementation is presented below.

We remark that our approach for avoiding breakdown is related to but different from the technique for the nonsymmetric Lanczos algorithm presented in [19]. In [19] a new Krylov subspace is appended to the existing Krylov subspace $\mathcal{K}_N$ and both subspaces are expanded after breakdown. In the present paper, we instead append a new Krylov subspace to $\mathcal{K}_{N-1}$ without further expanding $\mathcal{K}_{N-1}$ as follows. Let v_j denote the jth column of the matrix V_k in (1.2); i.e., $V_k = [v_1, v_2, \ldots, v_k]$. It follows from (1.3) that if $f_k \neq 0$, then $v_{k+1} = f_k/\|f_k\|$. Moreover, let $h_{i,j}$ denote the entry in position (i, j) of the matrices H_k in (1.2) or $\hat{H}_k$ in (1.4). It follows from (1.3) that $h_{k+1,k} = \|f_k\|$. Identifying the kth column of the right-hand and left-hand sides of (1.4) yields

$$Av_k = h_{1,k}v_1 + \cdots + h_{k,k}v_k + h_{k+1,k}v_{k+1}. \tag{3.1}$$

Assume that GMRES breaks down at step N. Then the Arnoldi decomposition (1.2) holds with $k = N$ and $f_N = 0$. If H_N is nonsingular, then GMRES finds the solution of (1.1) at this step. On the other hand, if H_N is singular, then GMRES cannot determine the solution of (1.1).

We remark that an exact breakdown is rare in actual computations in floating-point arithmetic. However, we have to be concerned about near-breakdown when the orthogonal projection of the vector Av_N into the complement of $\mathcal{K}_N$ is nonvanishing but "tiny." In this situation, we can still determine the last column $v_{N+1} = f_{N+1}/\|f_{N+1}\|$ of the matrix V_{N+1} in the Arnoldi decomposition (1.4), but the entry $h_{N+1,N}$ of $\hat{H}_N$ is tiny. If the matrix $\hat{H}_N$ is well conditioned, then this is a benign near-breakdown, and the computations can be continued with (standard) GMRES. On the other hand, if $\hat{H}_N$ is severely ill conditioned, then we are suffering from a hard near-breakdown, and standard GMRES will have difficulties finding a solution. Theorem 2.1 shows that in case of a hard breakdown at step N, with $A^N b \neq 0$, a component of the solution belongs to $\text{span}\{v_1, v_2, \ldots, v_{N-1}\} = \mathcal{K}_{N-1}$, but the column v_N of V_N is not required to represent the solution. We therefore replace this column by a unit vector, say, $\hat{v}$, which is orthogonal to the columns $v_1, v_2, \ldots, v_N$ of V_N. Such a vector can be generated, for example, by orthogonalizing a random vector against

$v_1, v_2, \ldots, v_N$. We also report numerical experiments in which we determine the vector $\hat{v}$ by orthogonalizing $A^T r_{N-1}$ against $v_1, v_2, \ldots, v_N$, where $r_{N-1} := b - Ax_{N-1}$ denotes the residual vector associated with the approximate solution x_{N-1} of (1.1). There are two reasons for the latter choice of $\hat{v}$. The vector $A^T r_{N-1}$ is parallel to the steepest descent direction for the functional

$$z \to \|Az - r_{N-1}\|^2,$$

and, therefore, we expect $\hat{v}$ to give rapid decrease of the norm of the residual error. Moreover, $A^T r_{N-1}$ is the residual error for the normal equations associated with (1.1). It may be pertinent to evaluate the residual error for the normal equations regularly when the linear system of equations (1.1) is inconsistent, because when this error is small, an acceptable approximate solution of (1.1) may have been found. The main disadvantage of this choice of $\hat{v}$ is that it requires a matrix-vector product evaluation with A^T. We therefore also present numerical examples when $\hat{v}$ is determined by orthogonalization of a random vector.

When we replace the last column of V_N by $\hat{v}$, i.e., $v_N := \hat{v}$, we create a new matrix U_1 and put the old v_N there. Specifically, we let v_N be the first column, denoted by u_1, of the matrix U_1, i.e., $U_1 = [u_1]$. Thus, analogously to (3.1), we obtain

$$Av_{N-1} = h_{1,N-1}v_1 + \cdots + h_{N-1,N-1}v_{N-1} + h_{N,N-1}u_1.$$

We can now compute a generalized Arnoldi decomposition, where, for $k := N, N+1, N+2, \ldots$, until the next breakdown occurs, we require the columns V_k to be orthonormal, as well as orthogonal to U_1. Thus, we obtain, for $k := N, N+1, N+2, \ldots$, until another breakdown occurs,

$$Av_k - V_k(V_k^T Av_k) - U_1(U_1^T Av_k) = f_k.$$

If a new near-breakdown occurs, say, if the entry $h_{k+1,k} = \|f_k\|$ of $\hat{H}_k$ is tiny, then the vector v_k is appended to the matrix U_1; i.e., we define $U_2 := [U_1, v_k]$, and a new unit vector, denoted by $\hat{v}$, which is orthogonal to all the columns of V_{k-1} and U_2, is generated. We replace the last column of V_k by $\hat{v}$, and the computations are continued in a similar fashion as after the first near-breakdown. The following algorithm implements this process. The right-hand side vector b of (1.1) is not required to be of unit length in the algorithm. The vector h_k denotes the last column of the upper Hessenberg matrix H_k in (1.2). Further, $\hat{H}_{k-1}(k,:)$ denotes the kth row of the matrix $\hat{H}_{k-1}$. The function $\mathrm{cond}(\hat{H}_k)$ evaluates the condition number $\|\hat{H}_k\|\|\hat{H}_k^\dagger\|$ of the matrix $\hat{H}_k$. The condition number is defined to be infinite if $\hat{H}_k$ is not of full rank.

ALGORITHM 1 (Breakdown-Free GMRES (BFGMRES)).

Input: $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$ ($b \neq 0$); *tol (threshold for breakdown)*
Initialize: $v_1 := b/\|b\|$; $p := 0$; $V_1 := [v_1]$; $U_0 := [\,]$; $\hat{H}_0 := [\,]$; $G_0 := [\,]$;
for $k := 1, 2, \cdots$ **until** *convergence*
  $w := Av_k$;
  $h_k := V_k^T w$; $g_k := U_p^T w$;
  $w := w - V_k h_k - U_p g_k$;
  $h_{k+1,k} := \|w\|$;
  $\hat{H}_k := \begin{bmatrix} \hat{H}_{k-1} & h_k \\ 0 & h_{k+1,k} \end{bmatrix}$;
  if $\mathrm{cond}(\hat{H}_k) > 1/tol$ **then**
    $U_{p+1} := [U_p, v_k]$;

11 $G_{k-1} := \begin{bmatrix} G_{k-1} \\ \hat{H}_{k-1}(k,:) \end{bmatrix}$; $\hat{H}_{k-1}(k,:) := 0$;
12 *Let $\hat{v}$ be a unit vector, such that $V_{k-1}^T \hat{v} = 0$, $U_{p+1}^T \hat{v} = 0$;*
13 *Replace the last column v_k of V_k by $\hat{v}$, i.e., let $v_k := \hat{v}$;*
14 $w := Av_k$;
15 $h_k := V_k^T w$; $g_k := U_{p+1}^T w$;
16 $w := w - V_k h_k - U_{p+1} g_k$;
17 $h_{k+1,k} := \|w\|$;
18 $\hat{H}_k := \begin{bmatrix} \hat{H}_{k-1} & h_k \\ 0 & h_{k+1,k} \end{bmatrix}$;
19 **if** $\mathrm{cond}(\hat{H}_k) > 1/tol$ **then goto** *line* 12;
20 $p := p + 1$;
21 **endif**
22 $v_{k+1} := w/h_{k+1,k}$;
23 $V_{k+1} := [V_k, v_{k+1}]$;
24 **if** $p > 0$ **then** $G_k := [G_{k-1}, g_k]$ **else** $G_k := G_{k-1}$ **endif**
25 *Solve* $\min_{y \in \mathbb{R}^k} \left\| \begin{bmatrix} \hat{H}_k \\ G_k \end{bmatrix} y - \|b\| e_1 \right\|$ for y_k;
26 $x_k := V_k y_k$;
27 **endfor**

The following remarks provide some detailed explanations of the algorithm:

- Lines 4–7 and 22 describe the generalized Arnoldi process for generating the vector v_{k+1}. We orthogonalize w against the columns of V_k and, when the matrix U_p is not empty, against the columns of U_p as well. Lines 5–6 describe classical Gram–Schmidt orthogonalization; however, our implementation employs the modified Gram–Schmidt procedure. When a hard near-breakdown is detected in line 9, v_k, the last column of V_k, is appended to the matrix U_p to yield U_{p+1}. The matrices $\hat{H}_{k-1}$ and G_{k-1} are updated in line 11 to yield the generalized Arnoldi decomposition

$$AV_{k-1} = V_k \hat{H}_{k-1} + U_{p+1} G_{k-1}.$$

  A new vector $\hat{v}$, which replaces the column v_k of V_k, is generated in line 12. After one step of the generalized Arnoldi process (lines 14–18), we check in line 19 whether the vector $\hat{v}$ yields a hard near-breakdown, in which case it is replaced. Otherwise, lines 22–24 yield the generalized Arnoldi decomposition

$$AV_k = V_{k+1} \hat{H}_k + U_p G_k = [V_{k+1}, U_p] \tilde{H}_k,$$

  where

$$\tilde{H}_k := \begin{bmatrix} \hat{H}_k \\ G_k \end{bmatrix}.$$

  It is clear that the matrix $[V_{k+1}, U_p]$ has orthogonal columns.
- Lines 25–26 determine a solution that minimizes the residual error from $\mathcal{R}(V_k)$, analogously as in standard GMRES. Writing $x = V_k y$, we have

$$\begin{aligned} \|Ax - b\| &= \|[V_{k+1}, U_p] \tilde{H}_k y - \beta_0 [V_{k+1}, U_p] e_1\| \\ &= \|\hat{H}_k y - \beta_0 e_1\|, \end{aligned}$$

where $\beta_0 := \|b\|$. Thus,

$$\min_{x\in\mathcal{R}(V_k)} \|Ax-b\| = \min_{y\in\mathbb{R}^k} \|\hat{H}_k y - \beta_0 e_1\|.$$

- Line 9 shows a simple criterion for a hard near-breakdown. We consider having reached a hard near-breakdown when the condition number $\mathrm{cond}(\hat{H}_k)$ is larger than the threshold $1/tol$. Since $\hat{H}_{k-1}$ is a submatrix of $\hat{H}_k$, the condition number is an increasing function of k. This criterion works well when $\mathrm{cond}(\hat{H}_k)$ increases slowly for small values of k and then for some larger values of k increases rapidly. However, it is not always suitable when the condition number increases steadily to a large value as k increases, because then when $\mathrm{cond}(\hat{H}_k) > 1/tol$, also $\mathrm{cond}(\hat{H}_j) \approx 1/tol$ for $j \approx k$, and this can result in several consecutive near-breakdowns. To avoid this undesirable situation, we propose to reduce tol by a factor, say, 10^{-2}, when a near-breakdown is encountered; i.e., we replace line 9 by

$$\textbf{if } \mathrm{cond}(\hat{H}_k) > 10^{2p}/tol \textbf{ then}, \tag{3.2}$$

  where p is the number of hard near-breakdowns encountered so far during the computations. A further modification of line 9 is discussed below.
- What we have presented is a version of full GMRES in which the memory and computational cost increase quickly with the iteration. In practice, a restarted version, where the algorithm is restarted after a certain number of iterations, should be used.

Algorithm 1 searches for a solution outside the Krylov subspace K_{N-1} by introducing a new vector $\hat{v}$ when a hard near-breakdown is detected. For singular matrices A with a null space of small dimension, hard near-breakdowns often do not occur until $N \approx n$ steps of the generalized Arnoldi process have been carried out. However, the component of the solution x of (1.1) in the Krylov subspace $\mathcal{K}_{N-1}$ (i.e., $\hat{x}$ in Theorem 2.1) sometimes can be approximated well by a vector x_k in a Krylov subspace $\mathcal{K}_k$ of dimension $k \ll N-1$, and then it would be desirable to introduce the new vector $\hat{v}$ already after k steps of the generalized Arnoldi process. This situation may be difficult to detect, because the residual vector $r_k := b - Ax_k$ associated with x_k might not be of small norm. We have found that it can be advantageous to generate a new vector $\hat{v}$ when the iterates x_k converge very slowly. This can be achieved by replacing line 9 of Algorithm 1 by

$$\textbf{if } (\mathrm{cond}(\hat{H}_k) > 10^{2p}/tol) \textbf{ or } (\|x_k - x_{k-1}\| \le \eta\|x_k\|) \textbf{ then}, \tag{3.3}$$

where η is a small positive constant of the order of the stopping tolerance and p is the number of hard near-breakdowns encountered during the computations so far; cf. (3.2).

We conclude this section with some comments on RRGMRES. The kth iterate, x_k, determined by RRGMRES, satisfies

$$\|b - Ax_k\| = \min_{z\in A\mathcal{K}_k} \|b - Az\|, \qquad x_k \in A\mathcal{K}_k.$$

Thus, the iterate belongs to the range of A. Computed examples in [6] illustrate that RRGMRES sometimes can yield computed solutions of higher quality than GMRES. We are therefore interested in an algorithm for BFRRGMRES, a breakdown-free

variant of RRGMRES. Such an algorithm can be obtained by a very minor modification of Algorithm 1: The initialization $v_1 := b/\|b\|$ in line 2 should be replaced by $v_1 := Ab/\|Ab\|$ and the vector $\|b\|e_1$ in line 25 has to be replaced by $[V_{k+1}, U_p]^T b$.

The following theorem discusses the behavior of RRGMRES. We remark that the formulation of Theorem 3.3 in [5], which discusses related results, is incomplete. This has recently been pointed out by Cao and Wang [8].

THEOREM 3.1. *Let the matrix $A \in \mathbb{R}^{n\times n}$ be of rank $N < n$ and apply RRGMRES with initial approximate solution $x_0 := 0$ to the solution of* (1.1). *Assume that RRGMRES breaks down at step N. If $\dim(A^2\mathcal{K}_N) = N$, then RRGMRES determines a least-squares solution of* (1.1). *If, instead, $\dim(A^2\mathcal{K}_N) < N$, then* (1.1) *has a solution in $A\mathcal{K}_N + \mathcal{R}(A^T)$.*

Proof. Assume that $\dim(A^2\mathcal{K}_N) = N$. This case is discussed by Cao and Wang [8]. Our proof is similar to the proof of Theorem 2.3 of section 2 and we use the same notation.

The Arnoldi process applied by RRGMRES breaks down at step N and yields the decomposition

$$AV_N = V_N H_N, \qquad V_N e_1 = Ab/\|Ab\|. \tag{3.4}$$

It follows from (3.4) and $\dim(A^2\mathcal{K}_N) = N$ that the matrix AV_N is of full rank and, therefore, that H_N is nonsingular. Moreover, $\mathcal{R}(V_N) \subset \mathcal{R}(A)$, and since $\operatorname{rank}(V_N) = \operatorname{rank}(A)$, it follows that $\mathcal{R}(V_N) = \mathcal{R}(A)$. The vector $\hat{x} \in \mathbb{R}^n$ is a least-squares solution of (1.1) if and only if the associated residual error is orthogonal to $\mathcal{R}(A)$, i.e., if and only if

$$0 = V_N^T(A\hat{x} - b). \tag{3.5}$$

The linear system of equations

$$H_N y = V_N^T b \tag{3.6}$$

has a unique solution $\hat{y}$. It follows from (3.4) that $\hat{x} := V_N\hat{y}$ satisfies (3.5). Thus, $\hat{x}$ is a least-squares solution of (1.1). RRGMRES determines this solution. This is a benign breakdown.

We turn to the case when $\dim(A^2\mathcal{K}_N) < N$. It follows from this inequality and (3.4) that the upper Hessenberg H_N is singular. Similarly as above, $\mathcal{R}(V_N) = \mathcal{R}(A)$ and, therefore, $\hat{x} \in \mathbb{R}^n$ is a least-squares solution of (1.1) if and only if $\hat{x}$ satisfies (3.5). However, differently from the situation above, the system of equations (3.6) might not have a solution, since H_N is singular. We circumvent this problem by appending a column to H_N as follows. Since H_N has positive subdiagonal entries, $\operatorname{rank}(H_N) = N - 1$. Let $u \in \mathcal{N}(H_N^T)$ be of unit length and define $v := A^\dagger V_N u \in \mathcal{R}(A^T)$. It follows from $\mathcal{R}(V_N) = \mathcal{R}(A)$ that $Av = V_N u$ and, therefore, $V_N^T Av = u$. We seek a least-squares solution of the form

$$x = V_N y + v\eta, \qquad y \in \mathbb{R}^N, \qquad \eta \in \mathbb{R}. \tag{3.7}$$

Substituting this expression into (3.5) yields

$$0 = V_N^T(AV_N y + Av\eta - b) = H_N y + u\eta - V_N^T b = [H_N, u]\begin{bmatrix} y \\ \eta \end{bmatrix} - e_1. \tag{3.8}$$

Since the matrix $[H_N, u] \in \mathbb{R}^{N\times(N+1)}$ is of full rank, (3.8) has a solution $\{\hat{y}, \hat{\eta}\}$. Substituting $y := \hat{y}$ and $\eta := \hat{\eta}$ into (3.7) yields a least-squares solution of (1.1). ∎

Theorem 3.1 implies that RRGMRES can be applied to solve inconsistent linear least-squares problems of the form (2.21).

4. Numerical examples. This section presents a few computed examples. All computations were carried out on an HP UNIX workstation using MATLAB with about 15 significant decimal digits. The initial approximate solution in all examples is chosen to be $x_0 := 0$.

Example 4.1. Consider a rectangular matrix of the form

$$\tilde{A} = \begin{bmatrix} \tilde{A}_{11} & 0 \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix} \in \mathbb{R}^{n\times\ell}, \tag{4.1}$$

where $\tilde{A}_{11} \in \mathbb{R}^{(n-k)\times(\ell-k)}$ is the sum of a random lower triangular matrix and $10I$, and $\tilde{A}_{21} \in \mathbb{R}^{k\times(\ell-k)}$ and $\tilde{A}_{22} \in \mathbb{R}^{k\times k}$ are random matrices generated by the MATLAB function `rand`.

We use GMRES and BFGMRES to solve the column-padded linear system of equations

$$Ax = b, \qquad A := [\tilde{A},\ 0] \in \mathbb{R}^{n\times n}, \qquad x := \begin{bmatrix} \tilde{x} \\ 0 \end{bmatrix} \in \mathbb{R}^n, \tag{4.2}$$

where $\tilde{A}$ is defined by (4.1). It is easy to see that if $\tilde{A}_{22}$ is nonsingular and $n - \ell \geq k$, then A has k zero eigenvalues associated with Jordan blocks of size at least 2. We also apply the conjugate gradient method, using the CGLS implementation, to solve the normal equations,

$$A^T Ax = A^T b, \tag{4.3}$$

associated with (1.1).

For ease of notation we use the matrix A in (4.2) instead of $\tilde{A}$. Let $x_k \in \mathbb{R}^n$ denote the kth iterate determined by any one of the iterative methods considered. For consistent linear systems of equations (1.1), we plot the norm of the residual vectors,

$$r_k := b - Ax_k, \tag{4.4}$$

relative to the norm of r_0 for increasing values of k. When the linear system (1.1) is inconsistent, we instead plot the norm of the residual error associated with the normal equations (4.3),

$$\hat{r}_k := A^T b - A^T Ax_k, \tag{4.5}$$

relative to the norm of $A^T\hat{r}_0$, because $\|\hat{r}_k\|$ vanishes when x_k is a least-squares solution of (1.1), while $\|r_k\|$ does not.

We first consider a consistent linear system of equations (1.1) with A defined by (4.1) and $n := 1000$, $\ell := 700$, $k := 3$. Let the solution $\tilde{x} \in \mathbb{R}^\ell$ be a vector with random entries and define the right-hand side by $b := \tilde{A}\tilde{x}$.

We use the criterion (3.2) with $tol := 10^{-8}$ for detecting hard near-breakdowns in BFGMRES. The vector $\hat{v}$ chosen in line 12 of Algorithm 1 at every hard near-breakdown is determined by orthogonalizing a random vector against the columns of the available matrices V_{k-1} and U_{p+1}. Here and throughout, the iterations are terminated when the relative residual $\|r_k\|/\|r_0\|$ drops below 10^{-14} (or $\|\hat{r}_k\|/\|\hat{r}_0\| < 10^{-14}$ for inconsistent systems).

The left-hand side graphs of Figure 4.1 show BFGMRES (solid curve) to reduce the norm of the residual error (4.4) faster than GMRES (dashed curve). We mark the

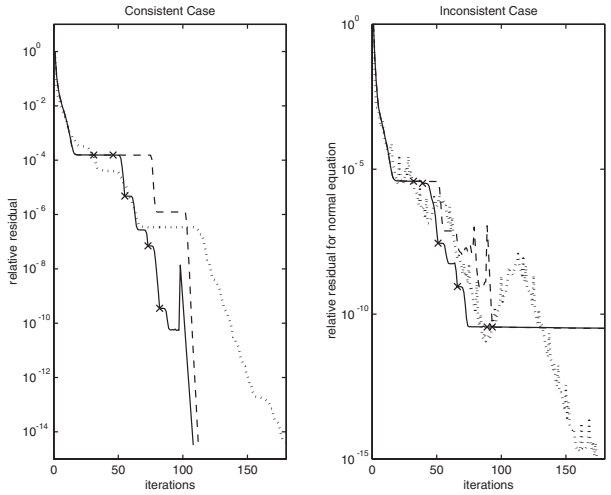


FIG. 4.1. *Example* 4.1: *Convergence histories for overdetermined linear system for BFGMRES (solid curves with near-breakdowns marked by* x*), GMRES (dashed curves), and CGLS (dotted curves).*

BFGMRES convergence curve by "x" where hard near-breakdowns occur. Here as well as in later examples, we see that there is typically significant reduction of the residual error a few iterations after a new vector is introduced. The nonmonotonic decrease of the norm of the residual error for BFGMRES near convergence is due to large round-off errors incurred when solving the reduced least-squares problem in line 25 of the algorithm. (Note that $\mathrm{cond}(\hat{H}_k)$ steadily increases and may become very large at that stage of iterations.) We also display the norm of the residual errors associated with iterates determined by CGLS (dotted curve). The latter method is seen to require more iterations than BFGMRES and GMRES to reduce the relative residual to 10^{-14}, but CGLS is implemented without reorthogonalization, and therefore requires less memory and fewer vector operations per iteration. On the other hand, note that CGLS needs the evaluation of two matrix-vector products in each iteration, one with A and one with A^T, while each iteration of BFGMRES or GMRES requires only the evaluation of one matrix-vector product with A.

We next illustrate the performance of BFGMRES, GMRES, and CGLS when applied to the solution of an inconsistent overdetermined linear system of equations. Such a system of equations is obtained by perturbing the right-hand side in (4.2). Specifically, we generate the right-hand side with the Matlab instruction `b=A*x+1e-6*rand(n,1)`, where $\mathtt{x}^T := [\tilde{x}^T, 0^T]$ and $\tilde{x} \in \mathbb{R}^\ell$ is the same random vector as above. The right-hand side graphs of Figure 4.1 show the norm of the residual errors (4.5) for the iterates x_k determined by GMRES, BFGMRES, and CGLS. GMRES and BFGMRES reduce the norm of the residual (4.5) by about a factor 10^{-11}, but thereafter the norm of the residual does not decrease further. BFGMRES converges somewhat faster than GMRES and gives a smoother reduction of the norm of the residual error. CGLS converges slower but is able to reduce the norm of the residual (4.5) by a factor 10^{-14}.

We turn to an underdetermined linear system of equations, obtained by using the transpose of the matrix A employed in the computations above. Thus, we would like

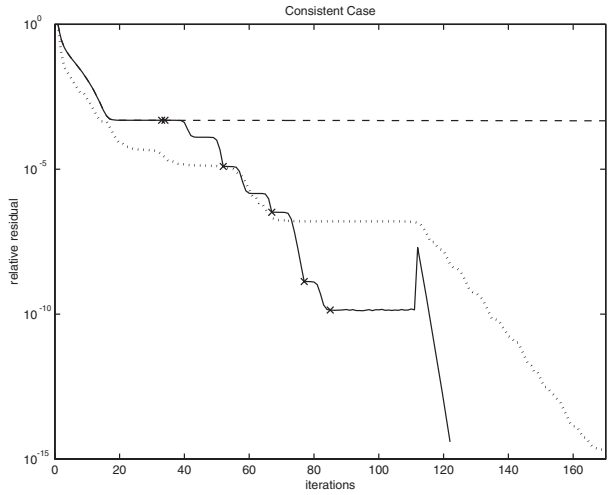


FIG. 4.2. *Example* 4.1: *Convergence histories for underdetermined linear system for BFGMRES (solid curve with near-breakdowns marked by* x*), GMRES (dashed curve), and CGLS (dotted curve).*

to solve

$$\tilde{A}^T y = \tilde{c},$$

where $y \in \mathbb{R}^n$ is a random vector and the right-hand side is defined by $\tilde{c} := \tilde{A}^T y$. BFGMRES and GMRES are applied to the associated row-padded system

$$A^T y = \begin{bmatrix} \tilde{c} \\ 0 \end{bmatrix}. \tag{4.6}$$

Figure 4.2 displays the norm of the residual error (4.4) for iterates computed by BFGMRES (solid curve), GMRES (dashed curve), and CGLS (dotted curve). Due to the structure of the linear system of equations (4.6), the last $n-m$ components of all vectors v_i determined by GMRES vanish. Therefore, GMRES cannot reduce the norm of the relative residual error below $5 \cdot 10^{-4}$. On the other hand, BFGMRES expands the subspace in which the computed solution is being sought, and the computed iterates converge to a solution of the linear system. Figure 4.2 shows BFGMRES to reduce the norm of the residual error (4.4) faster than any of the other methods considered.

Finally, we illustrate the finite termination property of Theorem 2.7 by solving an inconsistent overdetermined linear system of equations of the same form, but of smaller size, than the overdetermined system considered above. Thus, we solve an inconsistent system (4.2) with A of the form (4.1) with $n := 100$, $\ell := 70$, $k := 1$. The diagonal elements of A_{11} are $1, 2, \ldots, 69$. We compare the three methods both with and without reorthogonalization (for CGLS the residual vectors of the normal equations (4.5) are reorthogonalized). The graphs on the left-hand side of Figure 4.3 show the norm of the residual error for the normal equations converge for all three methods when reorthogonalization is carried out. The graphs show the relative residual norm to drop to 10^{-15} at step 71 (which is $N+1$) for GMRES and BFGMRES, but at step 70 for CGLS. This is consistent with the theory (Theorem 2.7). The graphs on the right-hand side of Figure 4.3 show the performance of the methods when no

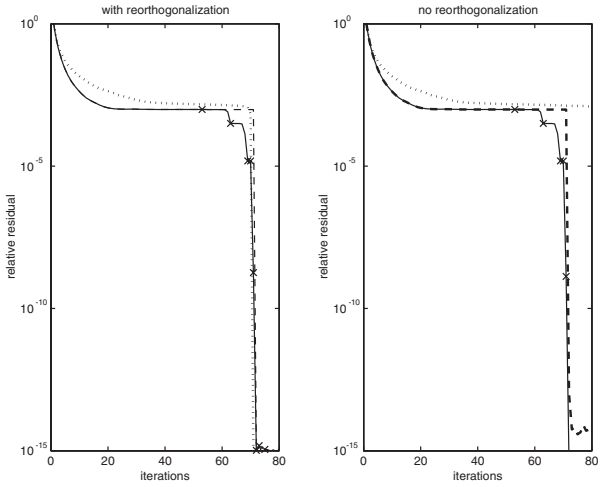


FIG. 4.3. *Example* 4.1: *Convergence histories for overdetermined linear system for BFGMRES (solid curves with near-breakdowns marked by* x*), GMRES (dashed curves), and CGLS (dotted curves).*

reorthogonalization has been carried out. GMRES and BFGMRES can be seen to behave similarly as with reorthogonalization, while iterates determined by CGLS do not appear to converge.

The next two examples are taken from Brown and Walker [4].

Example 4.2. Consider the nonsymmetric tridiagonal matrix

$$A := \begin{bmatrix} 0 & 1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & -1 & 0 \end{bmatrix} \in \mathbb{R}^{n\times n}. \tag{4.7}$$

For odd values of n, A is singular with index one. In particular, $\mathcal{N}(A) = \mathcal{N}(A^T)$. Let $n := 49$ and $b := [1, 0, \ldots, 0, 1]^T$ as in [4]. This yields an inconsistent linear system of equations (1.1). GMRES applied to this system computes a sequence of approximate solutions. At step 24 a hard near-breakdown is detected. Nevertheless, GMRES is able to determine a least-squares solution of the linear system of equations. The criterion used for detecting a near-breakdown is the same as in Example 4.1.

Now replace the right-hand side vector by $b := [1, 0, \ldots, 0, 1 + 10^{-10}]^T$. GMRES applied to (1.1) with this new right-hand side vector decreases the norm of the residual error of the associated normal equations (4.5) to 10^{-10} at step 24. However, in the next step the norm of the residual error increases by about a factor 10^4 due to high sensitivity to round-off errors. The residual error stays at this level for the next 25 iterations until it quickly drops to 10^{-14}. Figure 4.4 displays the convergence histories of the residual errors (4.5) for BFGMRES (solid curve) and GMRES (dashed curve). We see that BFGMRES yields a smoother decrease of the residual error than GMRES. The plateau in convergence of BFGMRES after step 24 appears to be due to the matrix itself as different choices of $\hat{v}$ result in similar convergence curves.

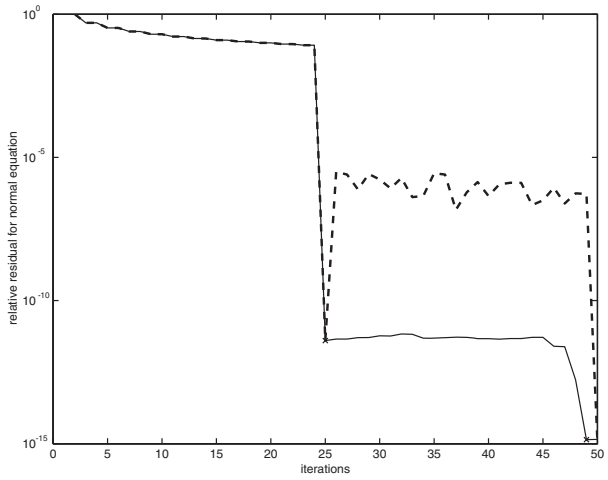


FIG. 4.4. *Example* 4.2: *Convergence histories for BFGMRES (solid curve with near-breakdowns marked by* x*) and GMRES (dashed curve).*

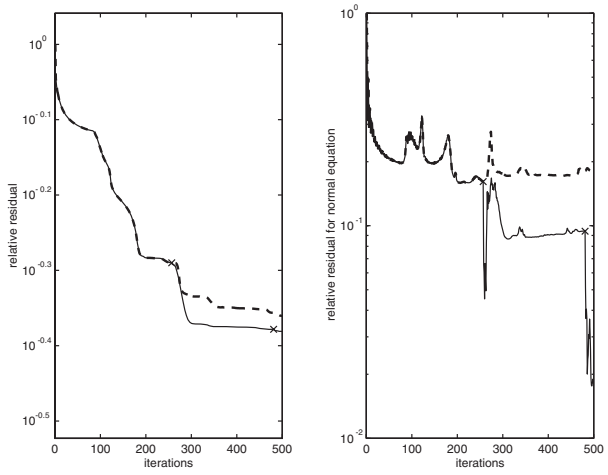


FIG. 4.5. *Example* 4.3: *Convergence histories for BFGMRES (solid curves with near-breakdowns marked by* x*) and GMRES (dashed curves).*

Example 4.3. We discretize the partial differential equation

$$\Delta u + d\frac{\partial u}{\partial z_1} = f, \qquad z := [z_1, z_2] \in [0,1]^2,$$

with Neumann boundary condition using centered finite differences on an $m \times m$ regular mesh; see [4, Experiment 4.3] for a description of the matrix so obtained. We used $m := 63$, $d := 10$, and

$$f(z) := z_1 + z_2 + \sin(10z_1)\cos(10z_2) + \exp(10z_1 z_2).$$

This yields an inconsistent linear system of equations which is fairly difficult to solve;

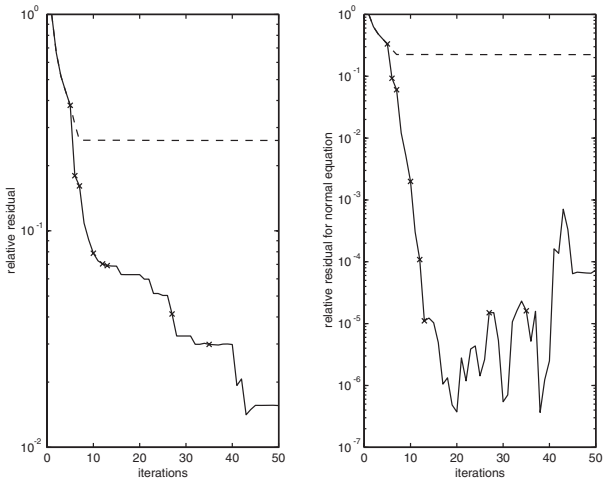


FIG. 4.6. *Example* 4.4: *Convergence histories for BFRRGMRES (solid curves with near-breakdowns marked by* x*) and RRGMRES (dashed curves).*

in particular, $\mathcal{N}(A^T) \neq \mathcal{N}(A)$. We use the criterion (3.3) with $tol := 10^{-8}$ and $\eta := 10^{-6}$. The vector $\hat{v}$ chosen in line 12 of Algorithm 1 at every hard near-breakdown is determined by orthogonalizing $A^T r_k$ against the columns of the available matrices V_{k-1} and U_{p+1}, where r_k denotes the residual vector (4.4) associated with the present iterate.

The convergence histories for GMRES and BFGMRES are shown in Figure 4.5. The condition numbers of the matrices $\hat{H}_k$ determined by GMRES increase steadily to about 10^{15} as k increases, while the condition numbers of the analogous matrices computed by BFGMRES are bounded by about 10^9. The smaller condition numbers associated with BFGMRES may help this method to achieve better convergence than GMRES.

Example 4.4. Our last example is generated by the Matlab function `parallax` in Regularization Tools by Hansen [13]. We used `parallax` to determine a rank-deficient matrix $\tilde{A} \in \mathbb{R}^{26\times 5000}$ and an associated right-hand side vector $b \in \mathbb{R}^{26}$. According to the Matlab function `rank`, the matrix $\tilde{A}$ has rank 24.

We apply RRGMRES and BFRRGMRES with full reorthogonalization to solve the underdetermined least-squares problem after row-padding of $\tilde{A}$. The graphs in the left-hand side of Figure 4.6 show the convergence histories of the norm of residual errors (4.4), and the graphs in the right-hand side of the figure display the convergence histories of the norm of residual errors of the normal equations (4.5) for BFRRGMRES (solid curves) and RRGMRES (dotted curves). The curves for BFRRGMRES vary less erratically than the curves for RRGMRES. Moreover, BFRRGMRES reduces the residual errors to a smaller value than RRGMRES.

In summary, our numerical examples demonstrate BFGMRES and BFRRGMRES to have more desirable convergence behavior than GMRES and RRGMRES, respectively, when applied to the solution of singular systems.

Acknowledgment. We would like to thank Bryan Lewis for discussions and comments.

REFERENCES

[1] O. Axelsson, *Conjugate gradient type methods for unsymmetric and inconsistent systems of linear equations*, Linear Algebra Appl., 29 (1980), pp. 1–16.

[2] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics Appl. Math. 9, SIAM, Philadelphia, 1994.

[3] Å. Björck, *Numerical Methods for Least-Squares Problems*, SIAM, Philadelphia, 1996.

[4] P. N. Brown and H. F. Walker, *GMRES on (nearly) singular systems*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 37–51.

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

[6] D. Calvetti, B. Lewis, and L. Reichel, *On the choice of subspace for iterative methods for linear discrete ill-posed problems*, Int. J. Appl. Math. Comput. Sci., 11 (2001), pp. 1069–1092.

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

[8] Z.-H. Cao and M. Wang, *A note on Krylov subspace methods for singular systems*, Linear Algebra Appl., 350 (2002), pp. 285–288.

[9] S. L. Campbell and C. D. Meyer, Jr., *Generalized Inverses of Linear Transformations*, Dover, New York, 1991.

[10] M. Eiermann, I. Marek, and W. Niethammer, *On the solution of singular linear systems of algebraic equations by semiiterative methods*, Numer. Math., 53 (1988), pp. 265–283.

[11] M. Eiermann and L. Reichel, *On the application of orthogonal polynomials to the iterative solution of singular systems of equations*, in Vector and Parallel Computing Issues in Applied Research and Development, J. Dongarra, I. Duff, P. Gaffney, and S. McKee, eds., Ellis Horwood, Chichester, UK, 1989, pp. 285–297.

[12] R. Freund and M. Hochbruck, *On the use of two QMR algorithms for solving singular systems and application in Markov chain modeling*, Numer. Linear Algebra Appl., 1 (1994), pp. 403–420.

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

[14] I. C. F. Ipsen and C. D. Meyer, *The idea behind Krylov methods*, Amer. Math. Monthly, 105 (1998), pp. 889–899.

[15] K. Kontovalis, R. J. Plemmons, and W. J. Stewart, *Block cyclic SOR for Markov chains with p-cyclic infinitesimal generator*, Linear Algebra Appl., 154/156 (1991), pp. 145–223.

[16] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.

[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] A. Sidi, *DGMRES: A GMRES-type algorithm for Drazin-inverse solution of singular nonsymmetric linear systems*, Linear Algebra Appl., 335 (2001), pp. 189–204.

[19] Q. Ye, *A breakdown-free variation of the nonsymmetric Lanczos algorithm*, Math. Comp., 62 (1994), pp. 179–207.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 4, pp. 1022–1057

A BIDIAGONAL MATRIX DETERMINES ITS HYPERBOLIC SVD TO VARIED RELATIVE ACCURACY*

BERESFORD N. PARLETT†

Abstract. Let $T = L\Omega L^t$ be an invertible, unreduced, indefinite tridiagonal symmetric matrix with Ω a diagonal signature matrix. We provide error bounds on the (relative) change in an eigenvalue and the angular change in its eigenvector when the entries in L suffer small relative changes. Our results extend those of Demmel and Kahan for $\Omega = I$. The relative condition number for an eigenvalue exceeds by 1 its absolute condition number as an eigenvalue of $\Omega L^t L$. The condition number of an eigenvector is a weighted sum of the relative separations of the eigenvalue from each of the others.

A small example shows that very small eigenpairs can be robust even when the large eigenvalues are extremely sensitive. When L is well conditioned for inversion, then all eigenvalues are robust and the eigenvectors depend only on the relative separations.

Key words. condition number, eigenvector, bidiagonal, hyperbolic SVD

AMS subject classifications. 15A18, 15A23

DOI. 10.1137/S0895479803424980

1. Setting the scene.

The first task is to provide a perspective from which both relative perturbation theory and hyperbolic singular values, described in the next section, are of interest. The underlying goal is the rapid and accurate calculation of the eigenvalues and eigenvectors of a real symmetric matrix. When the full set of eigenpairs is wanted, the computation is done in three phases: (1) reduction to real symmetric tridiagonal form T, (2) calculation of T's eigenpairs, and (3) the transformation of T's eigenvectors back to the given matrix. This paper is concerned entirely with the second phase.

Since the 1960s the QR algorithm has been used with excellent results for this phase. Its only defect is that QR always takes $O(n^3)$ arithmetic operations for an $n \times n$ matrix and, in principle, the job can be done with only $O(n^2)$ operations. One alternative to QR uses bisection for the eigenvalues and inverse iteration for the eigenvectors. This usually requires $O(n^2)$ effort but, sometimes, when there are large clusters of close eigenvalues that agree to three or more leading decimals, the LAPACK code DSTEIN, based on inverse iteration, slowed down significantly. An alternative LAPACK code DSBDC, based on the divide and conquer principle, takes between $O(n^2)$ and $O(n^3)$ operations, depending on the eigenvalue distribution, and does not adapt naturally to parallel implementation.

Even more recent is a method that can execute phase 2 in $O(n^2)$ operations even in the worst case. See [4], [5], and [6] for details. The key innovation in this approach is to replace T by several factorizations of the form $T - \sigma I = L\Omega L^t$, with L lower bidiagonal and $\Omega = \mathrm{diag}(\pm 1)$; each factorization represents a translate of T and hence has the same eigenvectors. When working in finite precision arithmetic it is essential that each factorization should determine its tiny eigenvalues to high relative accuracy. We say more about relative accuracy below but note here that for most symmetric matrices, including tridiagonal matrices, small relative changes in the

*Received by the editors March 13, 2003; accepted for publication (in revised form) by I.C.F. Ipsen June 11, 2004; published electronically May 6, 2005.
http://www.siam.org/journals/simax/26-4/42498.html
†Mathematics Department and Computer Science Division, EECS Department, University of California, Berkeley, CA 94720 (parlett@math.berkeley.edu).

matrix entries can provoke large relative changes in the eigenvalues near 0. It was this basic limitation that inspired us to look for a better representation for T and discover the virtues of $L\Omega L^t$. The hyperbolic singular values of L are just the square roots of the absolute values of the eigenvalues of $L\Omega L^t$. If these quantities are robust, then so are the eigenvalues themselves. In addition, our algorithm requires that the eigenvectors belonging to the tiny eigenvalues should also be accurately determined by L and Ω. That narrow investigation takes us into new territory and is the focus of this paper. We provide a bound, in Theorem 10.1, that reveals which eigenvectors of a possibly very ill-conditioned L are robust in the appropriate sense.

Before plunging into details we would like to make explicit two misgivings, or subliminal doubts, that can bother a specialist when initially exposed to relative perturbation theory for $L\Omega L^t$. The first reservation is that it is too much to expect small relative changes in tiny eigenvalues. There are 2×2 matrices for which tiny relative changes in an entry change the sign of the small eigenvalue. More generally, if there are several tiny eigenvalues that are close to each other it seems too good to be true that each of them would suffer only small relative changes. Backward stable algorithms, for symmetric matrices, give us eigenvalues with tiny errors, but tiny compared to the norm, not to the eigenvalue. Our response is that bidiagonal matrices are indeed special and L may determine each of $L\Omega L^t$'s tiny eigenvalues to this extreme accuracy, even when L is ill conditioned. When $\Omega = I$, then *all* the eigenvalues are robust in a relative sense. See the next paragraph for more on this topic. Moreover, when $L\Omega L^t$ is unreduced, then all its eigenvalues are simple and the eigenvectors may well be robust even when the eigenvalues are very close. The second reservation is that eigenvalue differences and eigenvectors are invariant under translation, and good taste demands that perturbation theory involve only quantities that are invariant (under translation). Our response is that, in contrast to T, $L\Omega L^t$ changes in a complicated way when it undergoes translation and it is this feature which makes relative perturbation theory not just reasonable but essential. In particular 0 eigenvalues are invariant even under large relative changes in L's entries. Here ends the big picture and we narrow the focus.

The primary concern of this paper is to determine how well L determines the spectral decomposition of $L\Omega L^t$. We begin with a little history. The celebrated result of Kahan (see [3]) says that when $\Omega = I$, then small *relative* changes in L's entries make only a small *relative* change in each eigenvalue even when L is ill conditioned. Later on, Deift, Demmel, Li, and Tomei [2] showed that the sensitivity of the eigenvectors is governed by the relative, not absolute, separation of the eigenvalues. Still later, simple proofs were found for both results; see [7], [8], and [11], [12], [10], respectively. This paper analyzes the problem in the case $\Omega \neq I$. It turns out that the same general scenario holds, but now the sensitivity of each eigenvalue is given by an amplification factor that may exceed 1 and the change in its eigenvector is amplified by a linear combination of the reciprocals of its relative separations from each of the other eigenvalues.

The hyperbolic singular value decomposition (HSVD) is not essential for presenting our analysis, but it is the most natural formulation of the extension of the earlier results to the case $\Omega \neq I$. The nice features of the standard SVD of a bidiagonal matrix extend, in many cases, to the hyperbolic singular values for arbitrary signature matrices Ω. One example is when the bidiagonal is well conditioned for inversion. The difficulty, in the ill-conditioned case, is that some of these singular values may be relatively robust while others are not. Even worse is the fact that the sensitivity

of the Ω singular vectors depends on the sensitivity of *all* the other Ω singular values. This raises the fear that the presence of some highly sensitive values could destabilize the Ω singular *vectors* of the robust values. In practice this fear appears to be unwarranted. What is needed are bounds that discriminate between robust and sensitive Ω singular triples, showing exactly how each relative gap contributes to the changes in a singular vector. The paper [14] anticipates the results given here but provides asymptotic estimates only, not bounds.

This paper presents discriminating bounds in Theorem 10.1. These bounds are needed in [6], [5] to justify claims for the numerical method mentioned above.

Related work. There has been a flowering of relative perturbation theory since the mid 1990s, helped by the biennial International Workshops on Accurate Solutions to Eigenvalue Problems (IWASEP). Papers [20], [16], [17], and [15] focus on our problem, namely, the precise way that the sensitivity of each eigenvalue affects the other eignvectors, and it would have been a great relief to have quoted the needed bounds from the literature and not written this paper. We suspect that, with their techniques, the authors of [16] and [17] could have obtained results close to our bounds had they been interested. However, the thrust of relative perturbation theory, until now, has been to obtain analogues of the big theorems in classical perturbation theory. In particular, the focus seems to be on conditions which ensure that all eigenvalues are determined to high relative accuracy by the data. In the four papers mentioned above all the results are of this type, and consequently the results on invariant subspaces depend only on the relative gap between the spectrum of the invariant subspace and its complement, as in the case of $\Omega = I$.

In contrast, in our applications the shifts are as close as possible to eigenvalue clusters, the L matrices are usually ill conditioned, and some eigenpairs are extremely sensitive. What to us is the most subtle and interesting aspect of these indefinite factorizations, as described above, is never even mentioned in [15], [17], and [20], as we now explain. While our singular value bound is no surprise and is implicit in the work of several authors (see [4], [16], and [17]), our eigenvector bound in Theorem 10.1 is realistic and new and explains how the destabilizing effect of extremely sensitive unwanted eigenvalues can be neutralized by the reciprocals of very large relative gaps, like $(1 - 10^{-8})/10^{-8}$, to produce robust eigenvectors for eigenvalues of $L\Omega L^t$ that are close to 0. The price to be paid for the realism is the detailed analysis of sections 7, 8, and 9.

Structure of paper. The analysis has two distinct phases. The first part, in sections 3 and 4, is well known and shows that for an unreduced, invertible, bidiagonal matrix L, small relative changes in the entries may be written in matrix form as $L \longrightarrow D_l L D_r$, where D_l and D_r are diagonal matrices close to I. The second part is new and, under mild conditions involving relative gaps, presents bounds on the changes in a singular triple $(\sigma, u, \Omega v)$ for *any* invertible matrix K under perturbations $K \longrightarrow D_l K D_r$ with D_l and D_r close to I and independent of K. When replacing bidiagonal L by general K, one must explicitly require that $K\Omega K^t$ have simple eigenvalues. The two-sided scaling is replaced by a diagonal congruence on a nonnormal matix of twice the size.

The analysis proper begins in section 5, which also contains a synopsis of the whole analysis, while section 6 gives the wanted error bounds but in terms of a matrix Z that solves a generalized Riccati equation derived in section 5. The rest of the paper is devoted to Z. Section 7 presents a slate of Sylvester operators on 2×2 matrices and the norms of their inverses. Section 8 contains the main theorem that includes

two bounds on $\|Z\|$. The first one is used to determine bounds on submatrices of Z which then combine to give the desired refined bound on $\|Z\|$. The first bound lurks only in higher order terms. Section 9 establishes the existence of Z under the same conditions needed in section 8. Section 10 inserts bounds from section 8 into those of section 6 to give our results and reveals the appropriate (relative) condition numbers. Section 11 shows how much the second bound on $\|Z\|$ improves on the first for certain extreme matrices.

Standard Householder notational conventions are used, including := for a definition and $\boldsymbol{v}^t$ for the transpose of $\boldsymbol{v}$, $\|\boldsymbol{v}\| := \sqrt{\boldsymbol{v}^t\boldsymbol{v}}$ for the Euclidean norm, $\|M\| := \sqrt{\lambda_{max}(M^tM)}$ for the spectral norm, and $\|M\|_F := \sqrt{\text{trace}(M^tM)}$ for the Frobenius norm, $\|\boldsymbol{v}\|_1 := \sum_i |v(i)|$.

2. The hyperbolic SVD. A diagonal matrix Ω whose diagonal entries are ± 1 is called a signature matrix. Given such a matrix Ω, the Ω-SVD (Ω-singular value decomposition) of a real invertible matrix K is a decomposition

$$K = U\Sigma V^t \tag{2.1}$$

where U is orthogonal, Σ is diagonal and positive definite, and V is Ω-orthogonal, and

$$V^t\Omega V = \hat{\Omega}, \quad \hat{\Omega} \text{ is another signature matrix.} \tag{2.2}$$

When Ω is indefinite, the Ω-SVD is said to be hyperbolic.

The decomposition (2.1)–(2.2) was introduced in [13] and extended in [1] to the case of rectangular matrices K, where delicate issues arise when $K\Omega K^t$ is rank deficient, but these issues do not arise in our invertible case. In [13] and [1] V is called hypernormal, but we prefer the term Ω-orthogonal. Some authors use $K = U\Sigma V^{-1}$ instead of (2.1).

By (2.2),

$$\hat{\Omega}V^t\Omega V = \hat{\Omega}(V^t\Omega V) = \hat{\Omega}^2 = I$$

and so

$$(\Omega V)^{-1} = \hat{\Omega}V^t. \tag{2.3}$$

Invoking (2.1) and (2.2) reveals that

$$\begin{aligned} K\Omega K^t &= U\Sigma(V^t\Omega V)\Sigma U^t \\ &= U(\Sigma^2\hat{\Omega})U^t. \end{aligned} \tag{2.4}$$

So U is an orthogonal eigenvector matrix of the real symmetric matrix $K\Omega K^t$ with eigenvalue matrix $\Lambda = \Sigma^2\hat{\Omega}$. The associated matrix

$$\begin{aligned} \Omega K^tK &= \Omega V\Sigma^2V^t \\ &= (\Omega V)\Sigma^2\hat{\Omega}(\hat{\Omega}V^t) \\ &= (\Omega V)\Lambda(\Omega V)^{-1} \quad \text{by (2.3).} \end{aligned} \tag{2.5}$$

Thus ΩV is an eigenvector matrix of the unsymmetric matrix ΩK^tK that is similar to $K\Omega K^t$. Since $(\Omega V)^{-1} = \hat{\Omega}V^t$, the similarity in (2.5) equalizes the spectral norms of the column and row eigenvector matrices of ΩK^tK. For a robust algorithm based on (2.1) and (2.4) see [15].

Note that for any conformable permutation matrix Π the transformation $U \longrightarrow U\Pi$, $V \longrightarrow V\Pi$ makes the simple reordering of (2.1)

$$K = (U\Pi)(\Pi^t\Omega\Pi)(V\Pi)^t,$$

whereas (2.2) becomes

$$(\Pi^t V^t)\Omega(V\Pi) = \Pi^t\hat{\Omega}\Pi.$$

So we may always order the eigenvalues in Λ so that $\hat{\Omega} = \Omega$, and this is what we do.

We make no use of the convention $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ often invoked for standard singular values. Even when Ω is indefinite, K^tK is positive definite, and so the pair (K^tK, Ω) is definite and its spectrum is real although ΩK^tK is not even normal.

Our results in section 10 involve $\|V\|_F$ so we now relate it to K.

LEMMA 2.1. *Let the Ω-SVD of invertible K be $U\Sigma V^t$. Then*

$$\|V^{-1}\|_F = \|V\|_F \leq \sqrt{\mathrm{cond}_F(K)} := \sqrt{\|K\|_F\,\|K^{-1}\|_F}. \tag{2.6}$$

Proof. Use $V^t\Omega V = \Omega$ and $U^t = U^{-1}$ to derive

$$K^tU = V\Sigma, \tag{2.7}$$

$$K^{-1}U = V^{-t}\Sigma^{-1} = (\Omega V\Omega)\Sigma^{-1}. \tag{2.8}$$

From (2.7),

$$U^tKK^tU = \Sigma V^tV\Sigma,$$

$$\mathrm{trace}\,[KK^t] = \sum_{i=1}^{n} \|\boldsymbol{v}_i\|^2\sigma_i^2. \tag{2.9}$$

From (2.8),

$$U^tK^{-t}K^{-1}U = \Sigma^{-1}(\Omega V\Omega)^t(\Omega V\Omega)\Sigma^{-1},$$

$$\mathrm{trace}\,[K^{-t}K^{-1}] = \sum_{i=1}^{n} \|\boldsymbol{v}_i\|^2\sigma_i^{-2}. \tag{2.10}$$

Finally,

$$\begin{aligned}
\|V\|_F^2 &= \sum_{i=1}^{n} \|\boldsymbol{v}_i\|^2 \\
&= \sum_{i=1}^{n} (\|\boldsymbol{v}_i\|\sigma_i)(\|\boldsymbol{v}_i\|\sigma_i^{-1}) \\
&\leq \left(\sum_{i=1}^{n} \|\boldsymbol{v}_i\|^2\sigma_i^2\right)^{1/2} \left(\sum_{i=1}^{n} \|\boldsymbol{v}_i\|^2\sigma_i^{-2}\right)^{1/2}, \quad \text{Cauchy–Schwarz,} \\
&= \mathrm{trace}\,[KK^t]^{1/2} \cdot \mathrm{trace}\,[K^{-t}K^{-1}]^{1/2} \quad \text{by (2.9) and (2.10),} \\
&= \|K\|_F\,\|K^{-1}\|_F = \mathrm{cond}_F(K). \qquad \square
\end{aligned}$$

A stronger result for the spectral norm was proved in [18]: $\|V^{-1}\| = \|V\| \leq \sqrt{\min \mathrm{cond}(KS)}$, the minimum over invertible S such that $S\Omega = \Omega S$.

3. Relative perturbations for bidiagonals. Let L be lower bidiagonal and $n \times n$:

$$L = \text{bidiag} \begin{pmatrix} \delta_1 & & \delta_2 & & \cdot & \cdot & & \delta_n \\ & l_1 & & l_2 & & \cdot & l_{n-1} & \end{pmatrix}.$$

Next we consider small relative perturbations to the $(2n-1)$ nontrivial entries of L: $l_i \longrightarrow l_i(1+\eta_i)$, $\delta_i \longrightarrow \delta_i(1+\varepsilon_i)$, where $|\eta_i| \le \varepsilon$, $|\varepsilon_i| \le \varepsilon$, and $\varepsilon \ll 1$ is called the level of perturbation. These changes may be represented in matrix form as

$$L \longrightarrow D_l L D_r \tag{3.1}$$

using diagonal scaling matrices D_l and D_r. These scaling matrices are not unique.

In order to avoid trivial cases we make two assumptions about L.

(H1) L is unreduced, no l_i vanishes, $i = 1, 2, \ldots, n-1$.

(H2) L is invertible, no δ_i vanishes, $i = 1, 2, \ldots, n$.

We call such L proper bidiagonals. Essentially there are two solutions for D_l and D_r:

$$\begin{aligned}
\text{Top Down:} \quad & D_l(1) = 1, \quad D_r(1) = (1+\varepsilon_1), \\
& D_l(k) = \prod_{i=1}^{k-1} \frac{1+\eta_i}{1+\varepsilon_i}, \quad k = 2, \ldots, n, \\
& D_r(k) = (1+\varepsilon_k)/D_l(k), \\
\text{Bottom Up:} \quad & D_r(n) = 1, \quad D_l(n) = (1+\varepsilon_n), \\
& D_r(k) = \prod_{i=k}^{n-1} \frac{1+\eta_i}{1+\varepsilon_{i+1}}, \quad k = n-1, \ldots, 1, \\
& D_l(k) = (1+\varepsilon_k)/D_r(k).
\end{aligned}$$

$$\text{Top Down:} \ \|D_l - I\| \le (1+\varepsilon)^{2n-2} - 1, \quad \|D_r - I\| \le (1+\varepsilon)^{2n-1} - 1,$$

$$\text{Bottom Up:} \ \|D_l - I\| \le (1+\varepsilon)^{2n-1} - 1, \quad \|D_r - I\| \le (1+\varepsilon)^{2n-2} - 1.$$

For later use we simplify these bounds by

$$\|D_l - I\| \le \varepsilon_d, \ \|D_l^{-1} - I\| \le \varepsilon_d, \ \|D_r - I\| \le \varepsilon_d, \ \|D_r^{-1} - I\| \le \varepsilon_d, \tag{3.2}$$

$$\varepsilon_d := (1+\varepsilon)^{2n-1} - 1.$$

We emphasize that D_l and D_r are independent of the entries of L.

Let the Ω-SVD of proper $n \times n$ L be $L = U\Sigma V^t$, with V satisfying (2.2). Note that $L^t \boldsymbol{u}_j = \boldsymbol{v}_j \sigma_j$, whereas $L(\Omega \boldsymbol{v}_j) = (\boldsymbol{u}_j \omega_j)\sigma_j$, and we say the Ω singular triples of L or L^t are $(\sigma_j, \boldsymbol{u}_j, \Omega \boldsymbol{v}_j)$, not $(\sigma_j, \boldsymbol{u}_j, \boldsymbol{v}_j)$, in order to keep Ω in view.

Let

$$D_l L D_r = \tilde{U}\tilde{\Sigma}\tilde{V}^t \tag{3.3}$$

be the Ω-SVD of the perturbed matrix.

This study gives bounds on $|\sigma_j - \tilde{\sigma_j}|/\sigma_j$, on $|\sin \angle(\boldsymbol{u}_j, \tilde{\boldsymbol{u}_j})|$, and on $|\sin \angle(\Omega \boldsymbol{v}_j, \Omega \tilde{\boldsymbol{v}_j})|$ for a typical j, $1 \le j \le n$. These bounds, given in Theorem 10.1, section 10, involve $\|\boldsymbol{v}_j\|$ and all the *relative* gaps $|\sigma_j - \sigma_k|/\sigma_j$, $k \ne j$.

4. The double matrix *B*. Bidiagonal form is essential for the bound (3.2) on D_l and D_r, but our bounds on the changes to the HSVD when L is perturbed to $D_l L D_r$ make no use of bidiagonal form. So we now replace L by any invertible K satisfying

$$\omega_j \sigma_j^2 \neq \omega_k \sigma_k^2, \quad k \neq j. \tag{4.1}$$

We follow the lead of Golub and Kahan [9] and others by turning an SVD problem into an eigenvalue problem of twice the size. If square K has Ω-SVD $K = U\Sigma V^t$, let

$$B := \begin{pmatrix} O & K \\ \Omega K^t & O \end{pmatrix}. \tag{4.2}$$

B is normal if and only if $\Omega = \pm I$. The case $\Omega = I$ was treated in the seminal paper [3]. In general, from (4.2) and (2.3),

$$\begin{aligned} B &= \begin{pmatrix} U & O \\ O & \Omega V \end{pmatrix} \begin{pmatrix} O & \Sigma \\ \Sigma & O \end{pmatrix} \begin{pmatrix} U^t & O \\ O & V^t \end{pmatrix} \\ &= \begin{pmatrix} U & O \\ O & \Omega V \end{pmatrix} \begin{pmatrix} O & \Sigma\Omega \\ \Sigma & O \end{pmatrix} \begin{pmatrix} U^{-1} & O \\ O & (\Omega V)^{-1} \end{pmatrix}. \end{aligned} \tag{4.3}$$

We use the following notation:

$$\begin{aligned} &U = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n], \quad V = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n], \\ &\Omega = \operatorname{diag}(\omega_1, \ldots, \omega_n), \quad \omega_i = \pm 1. \end{aligned}$$

It is convenient to work with a real (block) spectral decomposition of B, so we introduce the quantities that dominate the rest of the paper:

$$\begin{aligned} X &:= [X_1, \ldots, X_n], \qquad X_i = \begin{bmatrix} \boldsymbol{u}_i & \boldsymbol{o} \\ \boldsymbol{o} & \Omega \boldsymbol{v}_i \end{bmatrix}, \\ Y &:= [Y_1, \ldots, Y_n], \qquad Y_i = \begin{bmatrix} \boldsymbol{u}_i & \boldsymbol{o} \\ \boldsymbol{o} & \boldsymbol{v}_i \omega_i \end{bmatrix}. \end{aligned} \tag{4.4}$$

Both X and Y are orthogonal with respect to $I \oplus \Omega$. Note the subtle difference between Y_i and X_i. Using (2.3), (2.4), and (2.5), we list elementary properties used below:

$$\begin{aligned} &Y_i^t X_i = \operatorname{diag}(1, \omega_i \boldsymbol{v}_i^t \Omega \boldsymbol{v}_i) = \operatorname{diag}(1, \omega_i^2) = I_2, \\ &Y_i^t X_k = O, \quad k \neq i. \end{aligned}$$

So

$$Y^t X = I_{2n} \tag{4.5}$$

and, in addition,

$$Y_i^t Y_i = X_i^t X_i = \operatorname{diag}(1, \|\boldsymbol{v}_i\|^2). \tag{4.6}$$

Because $|\boldsymbol{v}_i^t \Omega \boldsymbol{v}_i| = |\omega_i| = 1$,

$$\|\boldsymbol{v}_i\|^2 \geq 1, \quad i = 1, \ldots, n. \tag{4.7}$$

Since $Y^t = X^{-1}$, $I_{2n} = XY^t = \sum_{i=1}^n X_i Y_i^t$.

By a suitable internal permutation of (4.3),

$$B = \sum_{i=1}^{n} \sigma_i X_i \Phi_i Y_i^t \tag{4.8}$$

with

$$\Phi_i = \begin{pmatrix} \boldsymbol{o} & \omega_i \\ 1 & \boldsymbol{o} \end{pmatrix}, \qquad \Phi_i^2 = \omega_i I_2. \tag{4.9}$$

Thus Φ_i distinguishes the real eigenvalues of B from the pure imaginary. From section 3 the assumption that L is proper guarantees

$$\sigma_i > 0, \;\; \sigma_i \neq \sigma_j, \;\; i \neq j, \;\; i = 1, \ldots, n. \tag{4.10}$$

We will work with the real spectral resolution of B given by (4.8). In the indefinite case, X and Y are not orthogonal and, using (4.5),

$$\begin{aligned} \|X_i Y_i^t\|^2 &= \lambda_{max}[X_i Y_i^t Y_i X_i^t] \\ &= \lambda_{max}[Y_i^t Y_i X_i^t X_i] \\ &= \lambda_{max}[\mathrm{diag}(1, \|\boldsymbol{v}_i\|^4)] = \|\boldsymbol{v}_i\|^4. \end{aligned} \tag{4.11}$$

By (4.5) the larger canonical angle $\alpha_i = \angle(\mathrm{range}\, X_i, \mathrm{range}\, Y_i)$ satisfies

$$\cos \alpha_i = \|\boldsymbol{v}_i\|^{-2}. \tag{4.12}$$

The smaller canonical angle is zero because $\boldsymbol{u}_i$ belongs to both spaces.

In the following sections we use (4.8) repeatedly in the form

$$BX_i = X_i \Phi_i \sigma_i, \quad i = 1, 2, \ldots, n. \tag{4.13}$$

In addition we will focus on a specific Ω singular triple $(\sigma_j, \boldsymbol{u}_j, \Omega \boldsymbol{v}_j)$ of K^t and we need a notation for those parts of matrices that do *not* involve j. We use the subscript $\langle j \rangle$ to denote this exclusion. Thus we redefine the X, Y of (4.4) by a harmless permutation

$$\begin{aligned} &X = [X_j, X_{\langle j \rangle}], \quad Y = [Y_j, Y_{\langle j \rangle}], \\ &\Phi = \Phi_j \oplus \Phi_{\langle j \rangle}, \quad \Sigma = \mathrm{diag}(\sigma_j, \Sigma_{\langle j \rangle}). \end{aligned} \tag{4.14}$$

In the later sections we also need

$$\widehat{\Sigma}_{\langle j \rangle} = \Sigma_{\langle j \rangle} \otimes I_2 = \mathrm{diag}(\sigma_1, \sigma_1, \sigma_2, \sigma_2, \ldots, \sigma_n, \sigma_n),$$

with the omission of σ_j.

5. Perturbation by diagonal congruence: The framework for the entire analysis. Observe that the perturbation $K \longrightarrow D_l K D_r$ described in section 2 corresponds to the perturbation $B \longrightarrow \tilde{B}$, where B is given in (4.2) and

$$\begin{aligned} \tilde{B} &= \begin{pmatrix} O & D_l K D_r \\ \Omega (D_l K D_r)^t & O \end{pmatrix}, \\ &= \begin{pmatrix} D_l & O \\ O & D_r \end{pmatrix} \begin{pmatrix} O & K \\ \Omega K^t & O \end{pmatrix} \begin{pmatrix} D_l & O \\ O & D_r \end{pmatrix}, \end{aligned}$$

because Ω and D_r commute. Thus we write

$$\tilde{B} = EBE \quad \text{with} \quad E = D_l \oplus D_r. \tag{5.1}$$

This simple diagonal congruence on B combined with (4.14) suggests that $E^{-1}X_j$ and $E^{-1}Y_j$ will yield small residuals since

$$R_j := (EBE)(E^{-1}X_j) - (E^{-1}X_j)\Phi_j\sigma_j = (E - E^{-1})X_j\Phi_j\sigma_j$$

and similarly for $E^{-1}Y_j$. Unfortunately, in the nonnormal case, small row and column residuals are not enough to bound the change in eigenvectors. We need global information. So we turn to other approaches.

Synopsis of proof. Although the details are complicated, the approach is standard and can be found in [19]. The framework is given in Lemma 5.1 below, the matrix N that block diagonalizes the perturbed matrix is given in (5.7), and the essential $(n-2)\times 2$ submatrix Z has to satisfy the Riccati equation (5.11). Given Z, the new $\tilde{u_j}$, $\Omega\tilde{v}$ vectors are given by (5.13) and Lemma 6.4. The new values $\tilde{\sigma_j}$ come from the Rayleigh quotient in (6.10). The proof of Lemma 6.3 is long because we want to get the higher order terms exactly. Sections 7, 8, and 9 are devoted to Z which is a column of (n-1) 2×2 diagonal matrices Z_{kj}, $k \neq j$. The Riccati equation (5.13) breaks down into $(n-1)$ Riccati equations, one for each Z_{kj}, $k \neq j$, given in (7.5). We solve this nonlinear equation by iteration and bound the mth iterate $Z_{kj}^{(m)}$ in (8.6).

This result (8.6) is vital to our realistic bound on the change in u_j because it contains the quotient $\|v_k\|^2/(|\sigma_k - \sigma_j|/\sigma_j)$ in which a huge numerator can be neutralized by a huge relative separation to give a small contribution to the perturbation. This was the reward for bounding each Z_{kj} carefully. Section 9 gives the conditions for convergence of the iteration, and section 10 substitutes the bound for Z into the results of section 6. The vector $\boldsymbol{m}_j$ in (10.7) and the eigenvector condition number (10.28) are the main original contributions of this paper.

Analysis. Write the spectral decomposition of $\tilde{B}$ as $\tilde{B} = \sum_{i=1}^{n} \tilde{\sigma}_i \tilde{X}_i \Phi_i \tilde{Y}_i^t$.

The way we obtain tight bounds on the sensitivity of an eigenvector u_j and a singular vector Ωv_j is by using a standard approach for a perturbed invariant subspace. So we rewrite (5.1) as an additive perturbation

$$\begin{aligned} \tilde{B} &= E^2(E^{-1}BE) \\ &= (I_{2n} + \mathcal{E}_2)E^{-1}BE, \end{aligned} \tag{5.2}$$

with

$$\mathcal{E}_2 := \operatorname{diag}(D_l^2 - I, D_r^2 - I).$$

The alternative formulation

$$\tilde{B} = B + \mathcal{E}_1 B + B\mathcal{E}_1 + \mathcal{E}_1 B \mathcal{E}_1,$$

for $\mathcal{E}_1 := E - I$, has three added terms, and the analysis is much more complicated than what follows.

LEMMA 5.1. *The perturbed matrix $\tilde{B} = EBE$ may be written in terms of the basis $E^{-1}X$, with inverse $(EY)^t$, as $\oplus_{i=1}^{n}\Phi_i\sigma_i + F = C + F$, using the notation of section 4. To focus on $\Phi_j\sigma_j$ this representation is written*

$$\tilde{C} = \begin{pmatrix} \Phi_j\sigma_j & O \\ O & \Phi_{\langle j\rangle}\widehat{\Sigma}_{\langle j\rangle} \end{pmatrix} + \begin{pmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{pmatrix},$$

$$\begin{pmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{pmatrix} = \begin{pmatrix} G_{11}\Phi_j\sigma_j & G_{12}\Phi_{\langle j\rangle}\widehat{\Sigma}_{\langle j\rangle} \\ G_{21}\Phi_j\sigma_j & G_{22}\Phi_{\langle j\rangle}\widehat{\Sigma}_{\langle j\rangle} \end{pmatrix}. \tag{5.3}$$

Each G matrix is an array of 2×2 diagonal matrices,

$$\begin{aligned} g_{ik} &= \operatorname{diag}(g^u_{ik}, \omega_i g^v_{ik}) \\ &:= \operatorname{diag}(\boldsymbol{u}_i^t(D_l^2 - I)\boldsymbol{u}_k, \omega_i \boldsymbol{v}_i^t(D_r^2 - I)\Omega\boldsymbol{v}_k), \end{aligned} \tag{5.4}$$

and $g^u_{ik} = g^u_{ki}$, $g^v_{ik} = g^v_{ki}$, $i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, n$.

Proof. From (4.8)

$$\begin{aligned} B &= XCY^t, \\ E^{-1}BE &= E^{-1}XCY^tE, \\ EBE &= E^2(E^{-1}BE) \\ &= E^{-1}BE + \mathcal{E}_2E^{-1}BE \\ &= (E^{-1}X)(C + F)(EY)^t, \end{aligned}$$

where

$$F := Y^tE\mathcal{E}_2E^{-1}XC = Y^t\mathcal{E}_2XC,$$

since E and $\mathcal{E}_2$ commute. To see the form of F, use the definitions

$$\begin{aligned} G_{11} &= Y_j^t\mathcal{E}_2X_j, \qquad G_{12} = Y_j^t\mathcal{E}_2X_{\langle j\rangle}, \\ G_{21} &= Y_{\langle j\rangle}^t\mathcal{E}_2X_j, \qquad G_{22} = Y_{\langle j\rangle}^t\mathcal{E}_2X_{\langle j\rangle}. \end{aligned}$$

The special form of X_i and Y_i is given in (4.4), and $\mathcal{E}_2$ is given in (5.2). The typical block in a G matrix is

$$\begin{aligned} g_{ik} &= Y_i^t\mathcal{E}_2X_k \\ &= \begin{pmatrix} \boldsymbol{u}_i & \boldsymbol{o} \\ \boldsymbol{o} & \boldsymbol{v}_i\omega_i \end{pmatrix}^t \begin{pmatrix} D_l^2 - I & O \\ O & D_r^2 - I \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_k & \boldsymbol{o} \\ \boldsymbol{o} & \Omega\boldsymbol{v}_k \end{pmatrix} \\ &= \operatorname{diag}(\boldsymbol{u}_i^t(D_l^2 - I)\boldsymbol{u}_k, \omega_i\boldsymbol{v}_i^t(D_r^2 - I)\Omega\boldsymbol{v}_k) \\ &= \operatorname{diag}(g^u_{ik}, \omega_i g^v_{ik}) \end{aligned}$$

defining the symmetric perturbation quantities g^u_{ik} and g^v_{ik}. ∎

Note that

$$G_{21} = \begin{pmatrix} g_{1j} \\ \vdots \\ g_{nj} \end{pmatrix}, \qquad G_{12} = (g_{j1}, \ldots, g_{jn}), \tag{5.5}$$

but with the g_{jj} term omitted. We chose the notation G_{22} rather than $G_{\langle j\rangle\langle j\rangle}$ despite the loss of precision. For application in the bounds to be developed in later sections, (5.4) and (4.7) yield

$$|g^u_{ik}| \le \|D_l^2 - I\|, \quad |g^v_{ik}| \le \|\boldsymbol{v}_i\|\,\|\boldsymbol{v}_k\|\,\|D_r^2 - I\|. \tag{5.6}$$

To obtain an invariant subspace of EBE that is close to range X_j we need to find a similarity transform of $\tilde{C} = C + F$ that annihilates the $(2,1)$ block. However,

we need to do more than that. The definition of X_j and Y_j in (4.4) shows that X_j determines Y_j. In the same way the new $\tilde{X}_j$ determines the new $\tilde{Y}_j$. So we must find a similarity transform that annihilates both the $(2,1)$ and $(1,2)$ blocks. A convenient notation is $C+F \longrightarrow N^{-1}(C+F)N$ with

$$N = \begin{pmatrix} I_2 & -\tilde{Z}^t \\ Z & I \end{pmatrix} \begin{pmatrix} Q_1 & O \\ O & Q_2 \end{pmatrix}, \quad N^{-1} = \begin{pmatrix} Q_1 & O \\ O & Q_2 \end{pmatrix} \begin{pmatrix} I_2 & \tilde{Z}^t \\ -Z & I \end{pmatrix}, \tag{5.7}$$

$$Q_1 = (I_2 + \tilde{Z}^t Z)^{-1/2}, \quad Q_2 = (I + Z\tilde{Z}^t)^{-1/2}, \tag{5.8}$$

where Z and $\tilde{Z}$ are to be chosen appropriately. Note that if

$$\begin{pmatrix} I_2 & \tilde{Z}^t \\ -Z & I \end{pmatrix} (\ C+F \ ) \begin{pmatrix} I_2 & -\tilde{Z}^t \\ Z & I \end{pmatrix} \tag{5.9}$$

is block diagonal, then so is $N^{-1}(C+F)N$ and therefore we may ignore Q_1 and Q_2 until later. Using $\tilde{C}$ as given in (5.3), the matrix in (5.9) is

$$\begin{bmatrix} \Phi_j\sigma_j + \tilde{Z}^t\Phi_{\langle j\rangle}\widehat{\Sigma}_{\langle j\rangle}Z, & -\Phi_j\sigma_j\tilde{Z}^t + \tilde{Z}^t\Phi_{\langle j\rangle}\widehat{\Sigma}_{\langle j\rangle} \\ -Z\Phi_j\sigma_j + \Phi_{\langle j\rangle}\widehat{\Sigma}_{\langle j\rangle}Z, & \Phi_{\langle j\rangle}\widehat{\Sigma}_{\langle j\rangle} + Z\Phi_{\langle j\rangle}\widehat{\Sigma}_{\langle j\rangle}\tilde{Z}^t \end{bmatrix} + \tag{5.10}$$

$$\begin{bmatrix} F_{11} + F_{12}Z + \tilde{Z}^t(F_{21} + F_{22}Z), & F_{12} - F_{11}\tilde{Z}^t + \tilde{Z}^tF_{22} - \tilde{Z}^tF_{21}\tilde{Z}^t \\ F_{21} - ZF_{11} + F_{22}Z - ZF_{12}Z, & F_{22} - F_{21}\tilde{Z}^t - Z(F_{12} - F_{11}\tilde{Z}^t) \end{bmatrix}.$$

Note that the $(2,1)$ block is a function of Z alone and the $(1,2)$ block is a function of $\tilde{Z}$ alone. It turns out that we do not need expressions for $\tilde{Z}$ because we can determine the perturbed eigenvalue $\tilde{\sigma}$ using a Rayleigh quotient instead of the $(1,1)$ block of (5.10).

From (5.10) $Z \in \mathbb{R}^{(2n-2)\times 2}$ must satisfy

$$Z\Phi_j\sigma_j - \Phi_{\langle j\rangle}\widehat{\Sigma}_{\langle j\rangle}Z = F_{21} - ZF_{11} + F_{22}Z - ZF_{12}Z. \tag{5.11}$$

This is called a generalized Riccati equation. In later sections we develop

- conditions for the existence of Z,
- the structure of Z,
- bounds on $\|Z\|$,
- bounds on each 2×2 block of Z.

A similar (long) analysis could be carried out for the generalized Riccati equation that corresponds to the $(1,2)$ block and determines $\tilde{Z}$. Fortunately we can replace this labor with Lemma 6.1 below because X_j and Y_j differ so little. Next we suppose that Z and $\tilde{Z}$ are in hand and so we have the similarity transformation by N that block diagonalizes $C+F$. From the proof of the lemma above we have

$$\tilde{B} = EBE = (E^{-1}XN)[N^{-1}(C+F)N][N^{-1}(EY)^t]. \tag{5.12}$$

From the proof of Lemma 5.1 and from N in (5.7), the invariant subspace range $(\tilde{X}_j)$ corresponding to range(X_j) has a basis $E^{-1}X\binom{I_2}{Z}Q_1$. For small enough Z and $\tilde{Z}$ the 2×2 matrix Q_1 is invertible and so we may ignore it in choosing a simple basis given by the columns of

$$\tilde{X}_j := E^{-1}X_j + E^{-1}X_{\langle j\rangle}Z. \tag{5.13}$$

We note that $\tilde{X}_j$ is not normalized in the same way as X_j. In section 8 we show that Z has the structure shown in (6.1), and so (5.13) splits into

$$\begin{aligned} \tilde{\boldsymbol{u}}_j &= D_l^{-1}(\boldsymbol{u}_j + U_{\langle j\rangle}\boldsymbol{z}_u), \\ \Omega\tilde{\boldsymbol{v}}_j &= D_r^{-1}(\Omega\boldsymbol{v}_j + \Omega V_{\langle j\rangle}\boldsymbol{z}_v), \end{aligned} \tag{5.14}$$

where $\boldsymbol{z}_u$ comes from the odd indices of column 1 of Z, $\boldsymbol{z}_v$ comes from the even indices of column 2 of Z, and both are used heavily in the next section.

6. Error bounds in terms of *Z*. This section exhibits the perturbed triple $(\tilde{\sigma}_j, \tilde{\boldsymbol{u}}_j, \Omega\tilde{\boldsymbol{v}}_j)$ when Z is known. We will show in Lemma 7.2 under mild conditions on the relative gaps among the $\{\sigma_i\}$ that a solution Z of (5.11) exists and is a tower of $(n-1)$ diagonal 2×2 matrices $Z_{1j}, Z_{2j}, \ldots, Z_{nj}$, with Z_{jj} omitted. In symbols,

$$Z = Z_{\langle j\rangle} = \begin{pmatrix} Z_{1j} \\ \vdots \\ Z_{nj} \end{pmatrix}, \qquad Z_{ij} \ \text{ diagonal.} \tag{6.1}$$

By definition $\|Z\|_F^2 = \sum_{k\neq j}\|Z_{kj}\|_F^2$, but, in general, $\|Z\|^2 \neq \sum_{k\neq j}\|Z_{kj}\|^2$ for the spectral norm. However, when the Z_{kj} are diagonal (or antidiagonal), there is a relation.

LEMMA 6.1. *If $Z = [\boldsymbol{z}_1, \boldsymbol{z}_2]$ has the structure given in* (6.1)*, then*

$$\|Z\|^2 \leq \sum_{k\neq j}\|Z_{kj}\|^2.$$

Proof. The two columns of Z have disjoint support and thus

$$\begin{aligned} Z^tZ &= \operatorname{diag}(\|\boldsymbol{z}_1\|^2, \|\boldsymbol{z}_2\|^2), \\ Z_{kj}^tZ_{kj} &= \operatorname{diag}(|z_1(2k-1)|^2, |z_2(2k)|^2). \end{aligned}$$

So

$$\begin{aligned} \|Z\|^2 &= \lambda_{max}(Z^tZ) \\ &= \max\left\{\sum_{k\neq j}|z_1(2k-1)|^2, \sum_{k\neq j}|z_2(2k)|^2\right\} \\ &\leq \sum_{k\neq j}\max\left\{|z_1(2k-1)|^2, |z_2(2k)|^2\right\} \\ &= \sum_{k\neq j}\|Z_{kj}\|^2. \qquad \square \end{aligned}$$

In the first draft of this paper we used the Frobenius norm. Now, with Lemma 6.1, we obtain tighter bounds.

Definition (5.13) shows how Z governs the relation of X_j to $\tilde{X}_j$. We also need the connection between Y_j and $\tilde{Y}_j$, which is messier than (5.13) but only through entry exchanges and sign changes. See Lemma 6.2.

Representation of $\tilde{Y}_j$. We need the following notation. Let $R := \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ and $R_+ = R \oplus \cdots \oplus R$, with $n-1$ terms.

LEMMA 6.2. *With the notation developed in the previous section we have* $\tilde{X}_j = E^{-1}(X_j + X_{\langle j\rangle}Z)$. *In addition* $\tilde{Y}_j$ *may be represented by* $E^{-1}(Y_j + Y_{\langle j\rangle}\bar{Z})$ *with*

$$\bar{Z} = R_+\Phi_{\langle j\rangle}ZR\Phi_j. \tag{6.2}$$

Note that $R_+\Phi_{\langle j\rangle} = \mathrm{diag}(1,\omega_1,1,\omega_2,\ldots,1,\omega_n)$, *with* $R\Phi_j = \mathrm{diag}\ (1,\omega_j)$ *omitted.*

Proof. Premultiply (5.14) by Ω and postmultiply by ω_j. Since ${D_r}^{-1}$ and Ω commute,

$$\begin{aligned}\tilde{\boldsymbol{v}}_j\omega_j &= {D_r}^{-1}(\boldsymbol{v}_j\omega_j + V_{\langle j\rangle}\boldsymbol{z}_v\omega_j)\\ &= {D_r}^{-1}(\boldsymbol{v}_j\omega_j + V_{\langle j\rangle}\Omega_{\langle j\rangle}\Omega_{\langle j\rangle}\boldsymbol{z}_v\omega_j).\end{aligned} \tag{6.3}$$

Whereas block k of Z is $\mathrm{diag}(z_u(k), z_v(k))$, the line above shows that block k of $\bar{Z}$ must be $\mathrm{diag}(z_u(k), \omega_k z_v(k)\omega_j)$. Now combine (6.3) with the expression for $\tilde{\boldsymbol{u}}_j$ in (5.14) to obtain

$$\tilde{Y}_j = \begin{pmatrix} \tilde{\boldsymbol{u}}_j & 0 \\ 0 & \tilde{\boldsymbol{v}}_j\omega_j \end{pmatrix} = E^{-1}(Y_j + Y_{\langle j\rangle}R_+\Phi_{\langle j\rangle}ZR\Phi_j). \qquad \square$$

An alternative representation of $\tilde{Y}_j$, coming from (5.7), is

$$\tilde{Y}_j = E(Y_j + Y_{\langle j\rangle}\tilde{Z}),$$

but we do not have a simple relation between $\tilde{Z}$ and Z, so we use Lemma 6.2 and $\bar{Z}$ instead. We apologize for the difficulty in distinguishing $\tilde{Z}$ from $\bar{Z}$, but from now on $\tilde{Z}$ drops out of the picture.

Our bounds involve the 2×2 matrix $\tilde{Y}_j^t\tilde{X}_j$. Although $\mathrm{range}(\tilde{X}_j)$ and $\mathrm{range}(\tilde{Y}_j)$ are the invariant subspaces of EBE associated with $\tilde{\sigma}_j$ close to σ_j, nevertheless $\tilde{X}_j$ and $\tilde{Y}_j$ are not properly normalized (i.e., they are not dual bases):

$$\tilde{Y}_j^t\tilde{X}_j = \begin{pmatrix} I_2 & \bar{Z}^t \end{pmatrix}(Y^tE^{-2}X)\begin{pmatrix} I_2 \\ Z \end{pmatrix} \tag{6.4}$$

$$= \begin{pmatrix} I_2 & \bar{Z}^t \end{pmatrix}(I+\bar{G})\begin{pmatrix} I_2 \\ Z \end{pmatrix}. \tag{6.5}$$

By (5.13) and (3.2), the bounds for the 2×2 blocks of $\bar{G}$ are identical to those of G since

$$\bar{G} = Y^t(E^{-2}-I)X \quad \text{and} \quad G = Y^t(E^2-I)X.$$

Change in σ_j. The long proof of Lemma 6.3 below reveals that Z only enters into the higher order terms of the bound. The dominant coefficient is $\|\boldsymbol{v}_j\|^2+1$. The Rayleigh quotient (matrix) of $\tilde{X}_j$ and $\tilde{Y}_j$ is the 2×2 antidiagonal matrix

$$\tilde{Y}_j^tEBE\tilde{X}_j(\tilde{Y}_j^t\tilde{X}_j)^{-1},$$

and so its eigenvalues are a $\pm$ pair. It represents the action of EBE on $\mathrm{range}(\tilde{X}_j)$, and its eigenvalues must be $\pm\sqrt{\omega_j\tilde{\sigma}_j^2} = \pm\sqrt{\tilde{\lambda}_j}$. To describe the higher order terms in the change to σ_j we need two new quantities,

$$\Upsilon_j := \sqrt{(n-1) + \|\boldsymbol{v}_j\|^2\,\|V_{\langle j\rangle}\|_F^2} \le \sqrt{2}\|\boldsymbol{v}_j\|\,\|V_{\langle j\rangle}\|_F, \tag{6.6}$$

$$\kappa_{\langle j\rangle} := \sigma_j^{-1}\sum_{i\neq j}\omega_i\sigma_i\det[Z_{ij}]. \tag{6.7}$$

It is more natural to state the bounds in Lemma 6.3 in terms of $\tilde{\lambda}_j$ and λ_j. Note that

$$\frac{|\tilde{\lambda}_j - \lambda_j|}{|\lambda_j|} = \frac{|\tilde{\sigma}_j - \sigma_j|}{\sigma_j} \cdot \frac{(\tilde{\sigma}_j + \sigma_j)}{\sigma_j} > \frac{|\tilde{\sigma}_j - \sigma_j|}{\sigma_j}.$$

In (3.2) we introduced ε_d, which bounds $\max(\|E - I\|, \|E^{-1} - I\|$, but in the analysis that follows the natural measure of the perturbation level is not ε_d but

$$\bar{\varepsilon}_d = \max(\|E^2 - I\|, \|E^{-2} - I\|) = \|\mathcal{E}_2\|$$

from (5.2). Reference to (3.2) shows that $\bar{\varepsilon}_d = \varepsilon_d(2 + \varepsilon_d)$.

LEMMA 6.3. *Provided that* $\|Z\|_F^2 \le \bar{\varepsilon}_d$,

$$\frac{|\tilde{\lambda}_j - \lambda_j|}{|\lambda_j|} \le \frac{\bar{\varepsilon}_d(\|\boldsymbol{v}_j\|^2 + 1) + \beta_2}{1 - \bar{\varepsilon}_d(\|\boldsymbol{v}_j\|^2 + 1) - \beta_2}, \tag{6.8}$$

$$\beta_2 := (\bar{\varepsilon}_d \Upsilon_j + \|Z\|_F)^2 + 2\kappa_{\langle j \rangle}. \tag{6.9}$$

Proof. We now begin a long and intricate calculation. By the properties of the Rayleigh quotient described above,

$$\tilde{\lambda}_j = -\det[\Phi_j \tilde{\sigma}_j] = -\det[\tilde{Y}_j^t EBE\tilde{X}_j (\tilde{Y}_j^t \tilde{X}_j)^{-1}].$$

Use Lemma 6.2 and (5.13) to write

$$\begin{aligned} \tilde{Y}_j^t EBE\tilde{X}_j &= \begin{pmatrix} I_2 & \bar{Z}^t \end{pmatrix} Y^t E^{-1}(EBE)E^{-1} X \begin{pmatrix} I_2 \\ Z \end{pmatrix} \\ &= \begin{pmatrix} I_2 & \bar{Z}^t \end{pmatrix} \Phi \widehat{\Sigma} \begin{pmatrix} I_2 \\ Z \end{pmatrix}. \end{aligned}$$

Invoke (6.5) for the inverse of $\tilde{Y}_j^t \tilde{X}_j$ to find

$$\tilde{\lambda_j} = -\frac{\det[\Phi_j \sigma_j + \bar{Z}^t \Phi_{\langle j \rangle} \widehat{\Sigma}_{\langle j \rangle} Z]}{\det\left[\begin{pmatrix} I_2 & \bar{Z}^t \end{pmatrix} (I + \bar{G}) \begin{pmatrix} I_2 \\ Z \end{pmatrix} \right]}. \tag{6.10}$$

Consider the numerator in (6.10) first. Note that $\Phi_{\langle j \rangle}^t R_+ = R_+ \Phi_{\langle j \rangle}$ and $\Phi_{\langle j \rangle}^2 = \widehat{\Omega}_{\langle j \rangle} = \Omega_{\langle j \rangle} \bigotimes I_2$. At the end of the following calculation we use the fact that Z_{ij} is diagonal to verify that ${Z_{ij}}^t R Z_{ij} = R\det[Z_{ij}]$. Lemma 6.2 gives $\bar{Z}$ and hence

$$\begin{aligned} \bar{Z}^t \Phi_{\langle j \rangle} \widehat{\Sigma}_{\langle j \rangle} Z &= \Phi_j^t R Z^t \Phi_{\langle j \rangle}^t R_+ \Phi_{\langle j \rangle} \widehat{\Sigma}_{\langle j \rangle} Z \\ &= \Phi_j^t R (Z^t R_+ \hat{\Omega}_{\langle j \rangle} \widehat{\Sigma}_{\langle j \rangle} Z) \\ &= \Phi_j^t R \sum_{i \ne j} \omega_i \sigma_i Z_{ij}^t R Z_{ij} \\ &= \Phi_j^t R \sum_{i \ne j} \omega_i \sigma_i R \det[Z_{ij}] \\ &= \Phi_{\langle j \rangle}^t \boldsymbol{\kappa}_{\langle j \rangle} \sigma_j, \end{aligned} \tag{6.11}$$

using (6.7). The denominator in (6.10) has more parts:

$$\begin{pmatrix} I_2 & \bar{Z}^t \end{pmatrix} (I + \bar{G}) \begin{pmatrix} I_2 \\ Z \end{pmatrix} = I_2 + \bar{Z}^t Z + (\bar{G}_{11} + \bar{G}_{12} Z + \bar{Z}^t \bar{G}_{21} + \bar{Z}^t \bar{G}_{22} Z),$$

and we bound each term separately. Use Lemma 6.2 again and

$$\bar{Z}_{ij}^t = \operatorname{diag}(\boldsymbol{z}_u(i), \omega_i \boldsymbol{z}_v(i)\omega_j), \quad Z_{ij} = \operatorname{diag}(\boldsymbol{z}_u(i), \boldsymbol{z}_v(i)),$$

to find that

$$\begin{aligned}
\bar{Z}^t Z &= \sum_{i\neq j} \bar{Z}_{ij}^t Z_{ij} \\
&= \operatorname{diag}\left(\sum_{i\neq j} \boldsymbol{z}_u(i)^2, \omega_j \sum_{i\neq j} \omega_i \boldsymbol{z}_v(i)^2 \right) \\
&= \operatorname{diag}\left(\|\boldsymbol{z}_u\|^2, \omega_j \boldsymbol{z}_v^t \Omega_{\langle j\rangle} \boldsymbol{z}_v \right).
\end{aligned}$$

Now we turn to the four terms in parentheses in the denominator of (6.10) that involve G. By the sentence after (6.5), the bounds on the blocks in $\bar{G}$ are the same as those in G. In order to understand the rest of the proof it is necessary to revisit (5.4), (5.5), and (5.6), where the 2×2 diagonal matrices $g_i k$ are revealed. The structure of Z is exhibited in (6.1) and Lemma 6.1. Finally, $\bar{\varepsilon}_d$ is given in (3.2):

$$\begin{aligned}
|\bar{G}_{11}| &\le \operatorname{diag}(\bar{\varepsilon}_d \|\boldsymbol{u}_j\|^2, \bar{\varepsilon}_d \|\boldsymbol{v}_j\|^2), \text{ the dominant term, from (5.6)}, \\
\bar{G}_{12} Z &= \sum_{i\neq j} \bar{G}_{ji} Z_{ij}, \\
|\bar{G}_{12} Z| &= \left| \sum_{i\neq j} \operatorname{diag}(\bar{g}_{ji}^u, w_j \bar{g}_{ji}^v) \operatorname{diag}(\boldsymbol{z}_u(i), \boldsymbol{z}_v(i)) \right| \\
&\le \bar{\varepsilon}_d \operatorname{diag}\left(\|\boldsymbol{z}_u\|_1, \|\boldsymbol{v}_j\| \sum_{i\neq j} \|\boldsymbol{v}_i\| \, |\boldsymbol{z}_v(i)| \right) \\
&\le \bar{\varepsilon}_d \operatorname{diag}(\|\boldsymbol{z}_u\|_1, \|\boldsymbol{v}_j\| \, \|V_{\langle j\rangle}\|_F \, \|\boldsymbol{z}_v\|), \text{ by Cauchy–Schwarz}.
\end{aligned}$$

Use (5.6) to see that the same bound holds for $|\bar{Z}^t \bar{G}_{21}|$. The quadratic term is

$$|\bar{Z}^t \bar{G}_{22} Z| = |\operatorname{diag}(\boldsymbol{z}_u^t \bar{G}_{22}^u \boldsymbol{z}_u, \omega_j \boldsymbol{z}_v^t \Omega_{\langle j\rangle} \bar{G}_{22}^v \boldsymbol{z}_v)|.$$

For the (i,k) block of $\bar{G}_{22}$ we have

$$\begin{aligned}
|(\bar{G}_{22}^u)_{ik}| &= |\boldsymbol{u}_i^t (D_l^{-2} - I)\boldsymbol{u}_k| \le \bar{\varepsilon}_d, \\
|(\bar{G}_{22}^v)_{ik}| &= |\omega_i \boldsymbol{v}_i^t (D_r^{-2} - I)\Omega \boldsymbol{v}_k| \le \bar{\varepsilon}_d \|\boldsymbol{v}_i\| \, \|\boldsymbol{v}_k\|.
\end{aligned}$$

So both matrices $|\bar{G}_{22}^u|$ and $|\bar{G}_{22}^v|$ are bounded, block by block, by rank-one matrices: the first has every entry $\bar{\varepsilon}_d$, and the second has (i,k) entry $\bar{\varepsilon}_d \|\boldsymbol{v}_i\| \|\boldsymbol{v}_k\|$. For the $(2,2)$ entry of $|\bar{Z}^t \bar{G}_{22} Z|$, by Cauchy–Schwarz,

$$|\boldsymbol{z}_v^t \Omega_{\langle j\rangle} \bar{G}_{22}^v \boldsymbol{z}_v| \le \bar{\varepsilon}_d \left(\sum_{i\neq j} \|\boldsymbol{v}_i\| \, |\boldsymbol{z}_v(i)| \right)^2 \le \bar{\varepsilon}_d (\|V_{\langle j\rangle}\|_F \, \|\boldsymbol{z}_v\|)^2,$$

and for the $(1,1)$ entry use $|x|$ for the vector with absolute values to find

$$|\boldsymbol{z}_u{}^t G_{22}{}^u \boldsymbol{z}_u| \le \bar{\varepsilon}_d [(1,\ldots,1)|\boldsymbol{z}_u|]^2 = \bar{\varepsilon}_d \|\boldsymbol{z}_u\|_1^2.$$

Thus, entry by entry,

$$|\bar{Z}^t \bar{G}_{22} Z| \le \bar{\varepsilon}_d \mathrm{diag}(\|\boldsymbol{z}_u\|_1^2, \|V_{\langle j\rangle}\|_F^2 \, \|\boldsymbol{z}_v\|^2).$$

Next assemble the six individual terms in the $(1,1)$ and $(2,2)$ entries of the 2×2 diagonal matrix $\begin{pmatrix} I_2 & \bar{Z}^t \end{pmatrix} (I + \bar{G}) \begin{pmatrix} I_2 \\ Z \end{pmatrix}$ to find

$$\begin{aligned}
\det \left[\begin{pmatrix} I_2 & \bar{Z}^t \end{pmatrix} (I + \bar{G}) \begin{pmatrix} I_2 \\ Z \end{pmatrix} \right] &\ge [1 - \bar{\varepsilon}_d - 2\bar{\varepsilon}_d \|\boldsymbol{z}_u\|_1 - \|\boldsymbol{z}_u\|^2 - \bar{\varepsilon}_d \|\boldsymbol{z}_u\|_1^2] \\
&\quad \cdot [1 - \bar{\varepsilon}_d \|\boldsymbol{v}_j\|^2 - 2\bar{\varepsilon}_d \|\boldsymbol{v}_j\| \|V_{\langle j\rangle}\|_F \|\boldsymbol{z}_v\| - \|\boldsymbol{z}_v\|^2 - \bar{\varepsilon}_d (\|V_{\langle j\rangle}\| \|\boldsymbol{z}_v\|)^2] \\
&= 1 - \bar{\varepsilon}_d (\|\boldsymbol{v}_j\|^2 + 1) - 2\bar{\varepsilon}_d (\|\boldsymbol{z}_u\|_1 + \|\boldsymbol{v}_j\| \, \|V_{\langle j\rangle}\|_F \, \|\boldsymbol{z}_v\|) \\
&\quad - \|Z\|_F^2 - \bar{\varepsilon}_d (\|\boldsymbol{z}_u\|_1^2 + \|\boldsymbol{z}_v\|^2 \, \|V_{\langle j\rangle}\|_F^2), \qquad (6.12)
\end{aligned}$$

ignoring positive products. By the Cauchy–Schwarz inequality,

$$\|\boldsymbol{z}_u\|_1 + \|\boldsymbol{v}_j\| \, \|V_{\langle j\rangle}\|_F \, \|\boldsymbol{z}_v\| \le \Upsilon_j \|Z\|_F,$$

with

$$\Upsilon_j^2 := (n-1) + \|\boldsymbol{v}_j\|^2 \, \|V_{\langle j\rangle}\|_F^2 \le 2\|\boldsymbol{v}_j\|^2 \, \|V_{\langle j\rangle}\|_F^2.$$

By subtracting and adding $(\bar{\varepsilon}_d \Upsilon_j)^2$ we can absorb the last term in (6.12),

$$\begin{aligned}
\det \left[\begin{pmatrix} I_2 & \bar{Z}^t \end{pmatrix} (I + \bar{G}) \begin{pmatrix} I_2 \\ Z \end{pmatrix} \right] &\ge 1 - \bar{\varepsilon}_d (\|\boldsymbol{v}_j\|^2 + 1) - (\bar{\varepsilon}_d \Upsilon_j + \|Z\|_F)^2 \\
&\quad + \bar{\varepsilon}_d^2 \Upsilon_j^2 - \bar{\varepsilon}_d \Upsilon_j^2 \|Z\|_F^2 \\
&\ge 1 - \bar{\varepsilon}_d (\|\boldsymbol{v}_j\|^2 + 1) - (\bar{\varepsilon}_d \Upsilon_j + \|Z\|_F)^2, \qquad (6.13)
\end{aligned}$$

invoking the hypothesis that $\|Z\|_F^2 \le \bar{\varepsilon}_d$. Insert (6.11) and (6.13) into (6.10) to find

$$\begin{aligned}
|\tilde{\lambda_j}| &\le \frac{|\omega_j \sigma_j (1 + \omega_j \boldsymbol{\kappa}_{\langle j\rangle}) \sigma_j (1 + \omega_j \boldsymbol{\kappa}_{\langle j\rangle})}{1 - \bar{\varepsilon}_d (\|\boldsymbol{v}_j\|^2 + 1) - (\bar{\varepsilon}_d \Upsilon_j + \|Z\|_F)^2} \\
&= \frac{|\lambda_j (1 + \omega_j \boldsymbol{\kappa}_{\langle j\rangle})^2|}{1 - \alpha - \beta^2},
\end{aligned}$$

defining α and β appropriately. Thus

$$\frac{|\tilde{\lambda_j} - \lambda_j|}{|\lambda_j|} \le \frac{2\boldsymbol{\kappa} + \boldsymbol{\kappa}^2 + \alpha + \beta^2}{1 - \alpha - \beta^2}.$$

Now $\boldsymbol{\kappa} = \boldsymbol{\kappa}_{\langle j\rangle}$ is quadratic in $\bar{\varepsilon}_d$ since it involves $\det[Z_{ij}]$. To eliminate the fourth-order term $\boldsymbol{\kappa}^2$ we modify the quadratic term in the denominator and write

$$\frac{2\boldsymbol{\kappa} + \boldsymbol{\kappa}^2 + \alpha + \beta^2}{1 - \alpha - \beta^2} \le \frac{\alpha + \beta^2 + 2\boldsymbol{\kappa}}{1 - (\alpha + \beta^2 + 2\boldsymbol{\kappa})},$$

and this yields the lemma with $\beta_2 := \beta^2 + 2\boldsymbol{\kappa}$. ∎

In section 10 we obtain bounds, under mild conditions, on $\|Z\|$ and $\|Z_{ij}\|$ and hence on $|\boldsymbol{\kappa}_{\langle j\rangle}|$. It turns out that both $\|Z\|_F^2$ and $|\boldsymbol{\kappa}_{\langle j\rangle}|$ are $O\left((\bar{\varepsilon}_d\|\boldsymbol{v}_j\|)^2\right)$, and hence the condition in Lemma 6.3, $\|Z\|_F^2 \le \bar{\varepsilon}_d$, is not restrictive. It follows that

$$\frac{|\tilde{\lambda}_j - \lambda_j|}{|\lambda_j|} \le 1 + 2(\|\boldsymbol{v}_j\|^2 + 1)\bar{\varepsilon}_d + O(\bar{\varepsilon}_d^2),$$

and $\|Z\|$ affects only the constant hidden by O but given in the lemma.

Change in u_j. Recall that $\tilde{X}_j = \begin{pmatrix} \tilde{\boldsymbol{u}}_j & \boldsymbol{o} \\ \boldsymbol{o} & \Omega\tilde{\boldsymbol{v}}_j \end{pmatrix}$.

LEMMA 6.4. *If Z satisfies* (5.11) *and* (6.1)*, then the unnormalized left Ω singular vector $\tilde{\boldsymbol{u}}_j$ belonging to $\tilde{\sigma}_j$ satisfies*

$$\begin{aligned} |\sin\angle(\tilde{\boldsymbol{u}}_j, \boldsymbol{u}_j)| &\le [\|\boldsymbol{z}_u\| + \|D_l^{-1} - I\|(1 + \|\boldsymbol{z}_u\|)](1 + \varepsilon_d) \\ &\le \frac{\|\boldsymbol{z}_u\| + \varepsilon_d}{(1 - \varepsilon_d)^2}. \end{aligned} \tag{6.14}$$

Proof. From (5.13), $\tilde{X}_j = E^{-1}X_j + E^{-1}X_{\langle j\rangle}Z$, and (6.1) guarantees that $\tilde{X}_j$ has the same form as X_j except that $\|\tilde{\boldsymbol{u}}_j\|$ need not be 1. The first column of (5.13) yields

$$\tilde{\boldsymbol{u}}_j = D_l^{-1}[\boldsymbol{u}_j + U_{\langle j\rangle}\boldsymbol{z}_u]. \tag{6.15}$$

Since U is orthogonal,

$$\begin{aligned} U_{\langle j\rangle}^t\tilde{\boldsymbol{u}}_j &= U_{\langle j\rangle}^t[\boldsymbol{u}_j + (D_l^{-1} - I)\boldsymbol{u}_j + U_{\langle j\rangle}\boldsymbol{z}_u + (D_l^{-1} - I)U_{\langle j\rangle}\boldsymbol{z}_u] \\ &= 0 + U_{\langle j\rangle}^t(D_l^{-1} - I)\boldsymbol{u}_j + \boldsymbol{z}_u + U_{\langle j\rangle}^t(D_l^{-1} - I)U_{\langle j\rangle}\boldsymbol{z}_u, \end{aligned}$$

and

$$1 \le \|\boldsymbol{u}_j + U_{\langle j\rangle}\boldsymbol{z}_u\| \le \|D_l\|\,\|\tilde{\boldsymbol{u}}_j\|.$$

Take norms and use the orthogonality of U to find

$$\begin{aligned} \|U_{\langle j\rangle}^t\tilde{\boldsymbol{u}}_j\| &\le \|D_l^{-1} - I\| + \|\boldsymbol{z}_u\|(1 + \|D_l^{-1} - I\|), \\ \|\tilde{\boldsymbol{u}}_j\| &\ge (1 + \varepsilon_d)^{-1} \ge 1 - \varepsilon_d, \end{aligned}$$

and (6.14) follows since $|\sin\angle(\tilde{\boldsymbol{u}}_j, \boldsymbol{u}_j)|$ is the quotient of the left-hand sides above. The second line in (6.14) absorbs the quadratic term by using an extra factor of $(1 - \varepsilon_d)$ in the denominator. ∎

Change in Ωv_j. ΩV is not orthogonal, so the bound on the change in $\Omega\boldsymbol{v}_j$ is more complicated than the bound on the change in $\boldsymbol{u}_j$. The quantity $\boldsymbol{\kappa}_j$ defined below is not related to the $\kappa_{\langle j\rangle}$ defined in (6.7); both are abbreviations for complicated expressions involving natural ingredients of the theory.

LEMMA 6.5. *If Z satisfies* (5.11) *and* (6.1)*, then the right Ω singular vector $\Omega\tilde{\boldsymbol{v}}_j$ for $\tilde{\sigma}_j$ satisfies*

$$|\sin\angle(\Omega\tilde{\boldsymbol{v}}_j, \Omega\boldsymbol{v}_j)| \le \left[\boldsymbol{\kappa}_j + \|D_r^{-1} - I\|(1 + \boldsymbol{\kappa}_j)\right](1 + \varepsilon_d\bar{\beta}_j), \tag{6.16}$$

$$\boldsymbol{\kappa}_j := \|V_{\langle j\rangle}\|\|\boldsymbol{z}_v\|\bar{\beta}_j,$$

where β_j is defined in (6.23) *below and satisfies*

$$\beta_j \le \bar{\beta}_j := \left(\|\boldsymbol{v}_j\| - \sqrt{\|\boldsymbol{v}_j\|^4 - 1}\, \|V_{\langle j\rangle}\| \|\boldsymbol{z}_v\| \right)^{-1}.$$

Proof. From (5.13), $\tilde{X}_j = E^{-1}X_j + E^{-1}X_{\langle j\rangle}Z$, and (6.1) guarantees that $\tilde{X}_j$ has the same form as X_j. The second column of $\tilde{X}_j$ has the form

$$\Omega\tilde{\boldsymbol{v}}_j = D_r^{-1}[\Omega\boldsymbol{v}_j + \Omega V_{\langle j\rangle}\boldsymbol{z}_v]. \tag{6.17}$$

The proof now follows the pattern used by Stewart in [19, p. 260]: If the columns of A_1 and A_2 are orthonormal bases for the spaces to be compared, then he completes them to $(A_1, B_1), (A_2, B_2)$ so that B_2 is orthogonal to A_1 and B_1 is orthogonal to A_2. For us, A_1 is $\Omega\boldsymbol{v}_j$ and A_2 is $\boldsymbol{v}_j\omega_j$. Let $\overset{o}{\boldsymbol{v}}_j = \boldsymbol{v}_j/\|\boldsymbol{v}_j\|$. In our case the natural complement to $\Omega\boldsymbol{v}_j$ is $Y_{\langle j\rangle} = V_{\langle j\rangle}\Omega_{\langle j\rangle}$ after normalization since $V_{\langle j\rangle}^t\Omega\boldsymbol{v}_j = 0$. Define

$$\begin{aligned} Q_{\langle j\rangle} &:= V_{\langle j\rangle}\Omega_{\langle j\rangle}(\Omega_{\langle j\rangle}V_{\langle j\rangle}^t V_{\langle j\rangle}\Omega_{\langle j\rangle})^{-1/2} \\ &= V_{\langle j\rangle}\Omega_{\langle j\rangle}K_{\langle j\rangle}, \quad \text{defining } K_{\langle j\rangle}. \end{aligned} \tag{6.18}$$

The matrices $(\Omega\boldsymbol{v}_j, \Omega V_{\langle j\rangle})$ and $(\boldsymbol{v}_j\omega_j, V_{\langle j\rangle}\Omega_{\langle j\rangle})$ are dual, i.e., $(V\Omega)^t(\Omega V) = \Omega V^t\Omega V = \Omega^2 = I$. Now we follow Stewart to determine the relations between $\overset{o}{\boldsymbol{v}}_j\ \omega_j, Q_{\langle j\rangle}$ and ΩV. We renormalize these matrices to preserve the ranges and duality, and incorporate $\Omega\ \overset{o}{\boldsymbol{v}}_j$ and $Q_{\langle j\rangle}$; thus

$$(\Omega\ \overset{o}{\boldsymbol{v}}_j, \Omega V_{\langle j\rangle}K_{\langle j\rangle}^{-1}) \quad \text{and} \quad (\overset{o}{\boldsymbol{v}}_j\ \omega_j\|\boldsymbol{v}_j\|^2, Q_{\langle j\rangle})$$

are still dual. By [19, Theorem 1.8] there is a unique vector $\boldsymbol{q}$ such that

$$\overset{o}{\boldsymbol{v}}_j\ \omega_j\|\boldsymbol{v}_j\|^2 = \Omega\ \overset{o}{\boldsymbol{v}}_j\ - Q_{\langle j\rangle}\boldsymbol{q}, \tag{6.19}$$

$$\Omega V_{\langle j\rangle}K_{\langle j\rangle}^{-1} = Q_{\langle j\rangle} + \Omega\ \overset{o}{\boldsymbol{v}}_j\ \boldsymbol{q}^t. \tag{6.20}$$

From (6.19), since $\|\Omega\ \overset{o}{\boldsymbol{v}}_j\ \| = 1$,

$$\|\boldsymbol{q}\| = \tan\angle(\overset{o}{\boldsymbol{v}}_j\ \omega_j, \Omega\ \overset{o}{\boldsymbol{v}}_j) = \sqrt{\|\boldsymbol{v}_j\|^4 - 1}, \tag{6.21}$$

and, after postmultiplying (6.20) by $K_{\langle j\rangle}$ and taking norms,

$$\|K_{\langle j\rangle}\| = \sqrt{\|\Omega V_{\langle j\rangle}\|^2 - \|\boldsymbol{q}^t K_{\langle j\rangle}\|^2} \le \|V_{\langle j\rangle}\|. \tag{6.22}$$

With (6.21) and (6.22) in hand, use (6.20) in (6.17) to find

$$\begin{aligned} \Omega\tilde{\boldsymbol{v}}_j &= D_r^{-1}[\Omega\ \overset{o}{\boldsymbol{v}}_j\ \|\boldsymbol{v}_j\| + (Q_{\langle j\rangle}K_{\langle j\rangle} + \Omega\ \overset{o}{\boldsymbol{v}}_j\ \boldsymbol{q}^t K_{\langle j\rangle})\boldsymbol{z}_v] \\ &= D_r^{-1}[\Omega\ \overset{o}{\boldsymbol{v}}_j\ \beta_j^{-1} + Q_{\langle j\rangle}K_{\langle j\rangle}\boldsymbol{z}_v], \end{aligned}$$

with

$$\beta_j := (\|\boldsymbol{v}_j\| + \boldsymbol{q}^t K_{\langle j\rangle}\boldsymbol{z}_v)^{-1}. \tag{6.23}$$

Multiply through by β_j to obtain a replacement for (6.17):

$$\Omega\tilde{\boldsymbol{v}}_j\beta_j = D_r^{-1}[\Omega\ \overset{o}{\boldsymbol{v}}_j + Q_{\langle j\rangle}K_{\langle j\rangle}\boldsymbol{z}_v\beta_j]. \tag{6.24}$$

Form the two components

$$\begin{aligned} Q_{\langle j\rangle}^t\Omega\tilde{\boldsymbol{v}}_j\beta_j &= 0 + Q_{\langle j\rangle}^t(D_r^{-1}-I)\Omega\ \overset{o}{\boldsymbol{v}}_j + K_{\langle j\rangle}\boldsymbol{z}_v\beta_j \\ &\quad + Q_{\langle j\rangle}^t(D_r^{-1}-I)Q_{\langle j\rangle}K_{\langle j\rangle}\boldsymbol{z}_v\beta_j, \\ 1 &\le \|\Omega\ \overset{o}{\boldsymbol{v}}_j + Q_{\langle j\rangle}K_{\langle j\rangle}\boldsymbol{z}_v\beta_j\| \le \|D_r\|\,\|\Omega\tilde{\boldsymbol{v}}_j\beta_j\|. \end{aligned}$$

Take norms and divide to get

$$\sin\angle(\Omega\tilde{\boldsymbol{v}}_j, \Omega\boldsymbol{v}_j) \le \left[\|K_{\langle j\rangle}\boldsymbol{z}_v\|\beta_j + \|D_r^{-1}-I\|(1+\|K_{\langle j\rangle}\boldsymbol{z}_v\|\beta_j)\right](1+\varepsilon_d\beta_j). \tag{6.25}$$

Use (6.21) and (6.22) in (6.23) and (6.25) and then (6.16) follows. ∎

7. The Sylvester operator S_{jk}. In the course of analyzing the generalized Riccati equation (5.11) we shall need some properties of the linear operator on $\mathbb{R}^{2\times 2}$ defined, for $k \ne j$, by

$$\begin{aligned} S_{jk}M &:= M\Phi_j\sigma_j - \Phi_k\sigma_k M \\ &= \begin{pmatrix} m_{12} & m_{11}\omega_j \\ m_{22} & m_{21}\omega_j \end{pmatrix}\sigma_j - \begin{pmatrix} m_{21}\omega_k & m_{22}\omega_k \\ m_{11} & m_{12} \end{pmatrix}\sigma_k \\ &= \begin{pmatrix} 0 & (\omega_j\sigma_j, -\omega_k\sigma_k)(m_{11}, m_{22})^t \\ (-\sigma_k, \sigma_j)(m_{11}, m_{22})^t & 0 \end{pmatrix} \\ &\quad + \begin{pmatrix} (\sigma_j, -\omega_k\sigma_k)(m_{12}, m_{21})^t & 0 \\ 0 & (-\sigma_k, \omega_j\sigma_j)(m_{12}, m_{21})^t \end{pmatrix}. \end{aligned} \tag{7.1}$$

The last equation shows that S_{jk} is a direct sum. The next result introduces the important quantities δ_{jk}.

LEMMA 7.1. *The operator S_{jk} is a direct sum of a mapping from diagonal matrices to antidiagonal matrices and another from antidiagonals to diagonals. By assumption* (4.1), *$\omega_k\sigma_k^2 \ne \omega_j\sigma_j^2$ and S_{jk} is invertible,*

$$S_{jk}^{-1}\begin{pmatrix} 0 & a \\ b & 0 \end{pmatrix} = \frac{1}{\omega_j\sigma_j^2 - \omega_k\sigma_k^2}\begin{bmatrix} (\sigma_j, \omega_k\sigma_k)(a,b)^t & 0 \\ 0 & (\sigma_k, \omega_j\sigma_j)(a,b)^t \end{bmatrix}, \tag{7.2}$$

$$S_{jk}^{-1}\begin{pmatrix} c & 0 \\ 0 & d \end{pmatrix} = \frac{1}{\omega_j\sigma_j^2 - \omega_k\sigma_k^2}\begin{bmatrix} 0 & (\omega_j\sigma_j, \omega_k\sigma_k)(c,d)^t \\ (\sigma_k, \sigma_j)(c,d)^t & \end{bmatrix}, \tag{7.3}$$

and

$$\delta_{jk}^{-1} := \|S_{jk}^{-1}\| = \begin{cases} |\sigma_j - \sigma_k|^{-1} & \text{if } \omega_k = \omega_j, \\ \dfrac{\sigma_j+\sigma_k}{\sigma_j^2+\sigma_k^2} & \text{if } \omega_k \ne \omega_j. \end{cases} \tag{7.4}$$

Proof. From (7.1) we may write $S_{jk} = S_{jk}^{\backslash} \oplus S_{jk}^{/}$. We consider $S_{jk}^{\backslash}$ acting on diagonal 2×2 matrices. In the appropriate coordinates

$$\begin{pmatrix} a \\ b \end{pmatrix} = S_{jk}^{\backslash}\begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} \omega_j\sigma_j & -\omega_k\sigma_k \\ -\sigma_k & \sigma_j \end{pmatrix}\begin{pmatrix} c \\ d \end{pmatrix}$$

so that

$$\begin{pmatrix} c \\ d \end{pmatrix} = \left(S_{jk}^{\backslash}\right)^{-1} \begin{pmatrix} a \\ b \end{pmatrix} = \frac{1}{\omega_j\sigma_j^2 - \omega_k\sigma_k^2} \begin{pmatrix} \sigma_j & \omega_k\sigma_k \\ \sigma_k & \omega_j\sigma_j \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}.$$

This gives (7.2). Moreover,

$$\left\| \begin{pmatrix} \sigma_j & \omega_k\sigma_k \\ \sigma_k & \omega_j\sigma_j \end{pmatrix} \right\|_\infty = \sigma_j + \sigma_k.$$

Finally, since $\left(S_{jk}^{\backslash}\right)^{-1}$ maps $\begin{pmatrix} 0 & a \\ b & 0 \end{pmatrix}$ into $\begin{pmatrix} c & 0 \\ 0 & d \end{pmatrix}$,

$$\begin{aligned} \left\|\left(S_{jk}^{\backslash}\right)^{-1}\right\| &= \max_{(a,b)} \left\| \begin{pmatrix} c & 0 \\ 0 & d \end{pmatrix} \right\| \Big/ \left\| \begin{pmatrix} 0 & a \\ b & 0 \end{pmatrix} \right\| \\ &= \max_{(a,b)} \left\| \begin{pmatrix} c \\ d \end{pmatrix} \right\|_\infty \Big/ \left\| \begin{pmatrix} a \\ b \end{pmatrix} \right\|_\infty \\ &= \begin{cases} \frac{\sigma_j+\sigma_k}{|\sigma_j^2-\sigma_k^2|} & \text{if } \omega_k = \omega_j, \\ \frac{\sigma_j+\sigma_k}{\sigma_j^2+\sigma_k^2} & \text{if } \omega_k \neq \omega_j, \end{cases} \end{aligned}$$

which yields (7.4).

The formula for $\left(S_{jk}^{/}\right)^{-1}$ is readily derived from (7.1), and $\left\|\left(S_{jk}^{/}\right)^{-1}\right\|$ is the same as (7.4). ∎

In our applications of Lemma 7.1 we use only $\left(S_{jk}^{\backslash}\right)^{-1}$ and, with little danger of confusion, we denote it by $(S_{jk})^{-1}$ with the appropriate restriction coming from the context. The quantities δ_{jk}^{-1} play a key role in the analysis that follows.

To see how S_{jk} comes into play, we write

$$Z = Z_{\langle j \rangle} = \begin{pmatrix} Z_{1j} \\ \vdots \\ Z_{nj} \end{pmatrix}$$

as in (6.1) but with no claims that Z_{kj} is diagonal. The kth (block) row of the Riccati equation (5.11) is $S_{jk}Z_{jk} = \mathcal{R}_k(Z)$ with

$$\begin{aligned} S_{jk}Z_{kj} &:= Z_{kj}\Phi_j\sigma_j - \Phi_k\sigma_k Z_{kj}, \\ \mathcal{R}_k(Z) &= g_{kj}\Phi_j\sigma_j - Z_{kj}g_{jj}\Phi_j\sigma_j + \sum_{i\neq j} g_{ki}\Phi_i\sigma_i Z_{ij} \\ &\quad - Z_{kj}\sum_{i\neq j} g_{ji}\Phi_i\sigma_i Z_{ij}. \end{aligned} \tag{7.5}$$

The 2×2 diagonal matrices g_{ik} are given in (5.4).

We will derive the structure of Z by using an iterative form of (7.5). Set $Z^{(0)} = O$ and, for each $k \neq j$, let

$$S_{jk}Z_{kj}^{(m+1)} = \mathcal{R}_k(Z^{(m)}), \quad m = 0, 1, 2, \ldots . \tag{7.6}$$

Set $m = 0$ in (7.6) to find that, for $k \neq j$,

$$S_{jk} Z_{kj}^{(1)} = g_{kj} \Phi_j \sigma_j.$$

From (5.4), $g_{kj}\Phi_j$ is antidiagonal and so, by Lemma 7.1,

$$Z_{kj}^{(1)} \text{ is diagonal, } k \neq j. \tag{7.7}$$

Moreover, by Lemma 7.1,

$$\begin{aligned} \left\| Z_{kj}^{(1)} \right\| &= \left\| {S_{jk}}^{-1} g_{kj} \Phi_j \sigma_j \right\| \\ &\leq \frac{\sigma_j}{\delta_{jk}} \left\| g_{kj} \Phi_j \right\| \qquad (7.8) \\ &= \frac{\sigma_j}{\delta_{jk}} \left\| g_{kj} \right\|, \qquad (7.9) \end{aligned}$$

and, by (5.4),

$$\left\| Z^{(1)} \right\|^2 \leq \sigma_j^2 \max \left\{ \sum_{k \neq j} \left(\frac{g_{kj}^u}{\delta_{jk}} \right)^2, \sum_{k \neq j} \left(\frac{g_{kj}^v}{\delta_{jk}} \right)^2 \right\}. \tag{7.10}$$

Although Z is much more complicated than $Z^{(1)}$, we shall find that $\|Z_{kj}\| \ \delta_{jk}/\sigma_j$ is bounded by something close to $\|g_{kj}\| = \max\{|g_{kj}^u|, |g_{kj}^v|\}$.

LEMMA 7.2. *For $m = 1, 2, \ldots$ the solutions $Z^{(m)}$ to* (7.6) *satisfy, for all $k \neq j$,*

$$Z_{kj}^{(m)} \text{ is diagonal.} \tag{7.11}$$

Proof. We proceed by induction. If each $Z_{kj}^{(m)}$ is diagonal, then each of the four terms in $\mathcal{R}_k(Z^{(m)})$ is antidiagonal: from (7.5), ignoring scalars σ_j,

$$\begin{aligned} g_{kj}\Phi_j &\text{ is (diag) (antidiag),} \\ Z_{kj}^{(m)} g_{jj} \Phi_j &\text{ is (diag) (diag) (antidiag),} \\ \sum_{i \neq j} g_{ki} \Phi_i Z_{ij}^{(m)} &\text{ is a sum of (diag) (antidiag) (diag),} \\ Z_{kj}^{(m)} \sum_{i \neq j} g_{ji} \Phi_i Z_{ij}^{(m)} &\text{ is diag } \sum \text{ (diag) (antidiag) (diag).} \end{aligned}$$

Since ${S_{jk}}^{-1}$ maps antidiagonals into diagonals,

$$Z_{kj}^{(m+1)} = {S_{jk}}^{-1} \mathcal{R}_k(Z^{(m)})$$

must be diagonal.

By (7.7), $Z_{kj}^{(1)}$ is diagonal for all $k \neq j$. Thus, by the finite induction principle, $Z_{kj}^{(m)}$ is diagonal for all positive integers m. ∎

8. A bound for $\|Z^{(m)}\|$ and $\|Z_{kj}^{(m)}\|$. First we obtain a simple bound on $\|Z^{(m)}\|$ and then use it to find a bound on $\|Z_{kj}^{(m)}\|$, which then gives rise to a better bound on $\|Z^{(m)}\|$. We regard (8.6)–(8.7) below as the principal technical contribution of this paper.

We shall need the following quantities:

$$\begin{aligned} \text{rgap} &:= \text{rgap}_j = \min_{k\neq j} \delta_{jk}/\sigma_j, \quad \delta_{jk} \text{ given in Lemma 7.1,} \\ \Delta_{\langle j\rangle} &:= \text{diag}(\delta_{j1}, \delta_{j1}, \delta_{j2}, \delta_{j2}, \ldots, \delta_{jn}, \delta_{jn}), \quad \delta_{jj} \text{ omitted,} \\ f_l &:= \left(\sum_{i\neq j} (\|g_{li}\|\sigma_i/\delta_{ji})^2 \right)^{1/2}. \end{aligned} \tag{8.1}$$

In order to explain the new quantities introduced in this section we invoke the diagonal form of each g_{ik} and Lemma 6.1 to obtain

$$\begin{aligned} \|G_{21}\| \leq \|\gamma^{(1)}\| &:= \left[\sum_{i\neq j} \|g_{ij}\|^2 \right]^{1/2}, \\ \|G_{12}\Phi_{\langle j\rangle}\widehat{\Sigma}_{\langle j\rangle}\Delta_{\langle j\rangle}^{-1}\| &\leq f_j, \\ \|G_{22}\Phi_{\langle j\rangle}\widehat{\Sigma}_{\langle j\rangle}\Delta_{\langle j\rangle}^{-1}\|^2 &\leq \sum_{i\neq j} f_i^2. \end{aligned}$$

The last matrix on the left may be permuted into a direct sum of two $(n-1)\times(n-1)$ submatrices, and the right side $\sum_{i\neq j} f_i^2$ bounds the Frobenius norm of each submatrix.

Consider the sequence $\{Z^{(m)}\}$ defined by (7.6).

THEOREM 8.1. *Assume that ε_d is small enough that the G matrices from* (5.3) *and* (5.4) *satisfy*

$$\|G_{11}\|/\text{rgap} \leq \frac{1}{4}, \tag{8.2}$$

$$\sum_{i\neq j} f_l^2 \leq \frac{1}{16}, \tag{8.3}$$

$$f_j(\|\gamma^{(1)}\|/\text{rgap}) \leq \frac{1}{32}; \tag{8.4}$$

then, for all $m \geq 1$,

$$\|Z^{(m)}\| \leq \frac{3\|\gamma^{(1)}\|}{\text{rgap}}, \tag{8.5}$$

and, for $k \neq j$,

$$\|Z_{kj}^{(m)}\| \leq \frac{\sigma_j}{\delta_{jk}}\tau_{kj}, \tag{8.6}$$

$$\tau_{kj} := \frac{\|g_{kj}\| + 3f_k\|\gamma^{(1)}\|}{1 - (\|G_{11}\| + 3f_j\|\gamma^{(1)}\|)/\text{rgap}_j}, \tag{8.7}$$

so that

$$\|Z^{(m)}\|^2 \le \sum_{k\neq j}\left(\frac{\sigma_j\tau_{kj}}{\delta_{jk}}\right)^2. \tag{8.8}$$

Proof. Recall (7.6):

$$S_{jk}Z_{kj}^{(m+1)} = g_{kj}\Phi_j\sigma_j - Z_{kj}^{(m)}G_{11}\Phi_j\sigma_j + \sum_{i\neq j} g_{ki}\Phi_i\sigma_i Z_{ij}^{(m)} - Z_{kj}^{(m)}G_{12}\Phi_{\langle j\rangle}\widehat{\Sigma}_{\langle j\rangle}Z^{(m)}.$$

Premultiply by S_{jk}^{-1} and take norms

$$\|Z_{kj}^{(m+1)}\| \le \delta_{jk}^{-1}\Bigg[\|g_{kj}\|\sigma_j + \|Z_{kj}^{(m)}\|\|G_{11}\|\sigma_j + \sum_{i\neq j}\|g_{ki}\|\sigma_i\|Z_{ij}^{(m)}\| + \|Z_{kj}^{(m)}\|\|G_{12}\Phi_{\langle j\rangle}\widehat{\Sigma}_{\langle j\rangle}Z^{(m)}\|\Bigg]. \tag{8.9}$$

Now define ${\gamma_i}^{(m)}$, hiding its dependence on j, by

$$\|Z_{ij}^{(m)}\| = \frac{\sigma_j}{\delta_{ji}}{\gamma_i}^{(m)}, \quad i\neq j, \tag{8.10}$$

$$\gamma^{(m)} := (\gamma_1^{(m)},\dots,\gamma_n^{(m)})^t, \quad \gamma_i^{(m)} \text{ omitted.}$$

Insert (8.10) into (8.9) to find

$$\|Z_{kj}^{(m+1)}\| \le \frac{\sigma_j}{\delta_{jk}}\Bigg[\|g_{kj}\| + \frac{\sigma_j}{\delta_{jk}}{\gamma_k}^{(m)}\|G_{11}\| + \sum_{i\neq j}\|g_{ki}\|\sigma_i\frac{{\gamma_i}^{(m)}}{\delta_{ji}} + \frac{\sigma_j}{\delta_{jk}}{\gamma_k}^{(m)}\sum_{i\neq j}\|g_{ji}\|\sigma_i\frac{{\gamma_i}^{(m)}}{\delta_{ji}}\Bigg]. \tag{8.11}$$

Apply Cauchy–Schwarz to each sum in (8.11) to isolate the ${\gamma_i}^{(m)}$. Divide (8.11) through by σ_j/δ_{jk} to find an inequality for ${\gamma_k}^{(m+1)}$ with the help of (8.1):

$${\gamma_k}^{(m+1)} \le \|g_{kj}\| + \frac{\|G_{11}\|}{\text{rgap}}{\gamma_k}^{(m)} + f_k\|\gamma^{(m)}\| + \frac{f_j}{\text{rgap}}{\gamma_k}^{(m)}\|\gamma^{(m)}\|. \tag{8.12}$$

Square (8.12) and invoke $(a+b+c+d)^2 \le 4(a^2+b^2+c^2+d^2)$ to obtain

$$\left({\gamma_k}^{(m+1)}\right)^2 \le 4\Bigg[\|g_{kj}\|^2 + \left(\frac{\|G_{11}\|}{\text{rgap}}\right)^2\left({\gamma_k}^{(m)}\right)^2 + f_k^2\|\gamma^{(m)}\|^2 + \left(\frac{f_j}{\text{rgap}}\right)^2\left({\gamma_k}^{(m)}\right)^2\|\gamma^{(m)}\|^2\Bigg].$$

Next sum over $k\neq j$, to find

$$\frac{1}{4}\left\|\gamma^{(m+1)}\right\|^2 \le \|\gamma^{(1)}\|^2 + \left[\left(\frac{\|G_{11}\|}{\text{rgap}}\right)^2 + \sum_{k\neq j} f_k^2\right]\|\gamma^{(m)}\|^2 + \left(\frac{f_j}{\text{rgap}}\right)^2\|\gamma^{(m)}\|^4. \tag{8.13}$$

Define the associated quadratic equation

$$\frac{1}{4}\xi^2 = \|\gamma^{(1)}\|^2 + \left[\left(\frac{\|G_{11}\|}{\text{rgap}}\right)^2 + \sum_{k\neq j} f_k^2\right]\xi^2 + \left(\frac{f_j}{\text{rgap}}\right)^2 \xi^4 \tag{8.14}$$
$$= Q(\xi^2).$$

From (8.1) we can see that the f's depend on complementary relative gaps δ_{jk}/σ_k, not δ_{jk}/σ_j. The hypotheses (8.2)–(8.4) guarantee that the smaller solution ξ_j^2 of $\xi^2 = 4Q(\xi^2)$ is positive and satisfies

$$\xi_j^2 = \frac{2\|\gamma^{(1)}\|^2}{\mathcal{A} + \sqrt{\mathcal{A}^2 - (2\|\gamma^{(1)}\|f_j/\text{rgap})^2}},$$

where

$$\mathcal{A} := \frac{1}{4} - \left(\frac{\|G_{11}\|}{\text{rgap}}\right)^2 - \sum_{k\neq j} f_k^2$$
$$\geq \frac{1}{8} \quad \text{by (8.2) and (8.3).} \tag{8.15}$$

Hypothesis (8.4) was chosen so that

$$\mathcal{A}^2 - (2\|\gamma^{(1)}\|f_j/\text{rgap})^2 \geq \left(\frac{1}{8}\right)^2 - \left(\frac{2}{32}\right)^2 = \frac{3}{16^2}.$$

Thus

$$4\|\gamma^{(1)}\|^2 \leq \xi_j^2 \leq \frac{32}{2+\sqrt{3}}\|\gamma^{(1)}\|^2,$$

and

$$2\|\gamma^{(1)}\| \leq \xi_j \leq 3\|\gamma^{(1)}\|. \tag{8.16}$$

Next we show, by induction, that $\|\gamma^{(m)}\| < \xi_j$. By (8.16), $\|\gamma^{(1)}\| \leq \xi_j$. In general if $\|\gamma^{(m)}\| < \xi_j$, then, by (8.13) and (8.14),

$$\|\gamma^{(m+1)}\|^2 \leq 4Q(\|\gamma^{(m)}\|^2) < 4Q(\xi_j^2) = \xi_j^2. \tag{8.17}$$

The second inequality uses the fact that all of Q's coefficients are positive. Thus (8.17) holds for all m. Now (8.16) and (8.17) yield the bound (8.5). Since $Z^{(m)}$'s columns are orthogonal, (8.10) yields

$$\|Z^{(m)}\|^2 \leq \sum_{k\neq j}\left(\frac{\sigma_j}{\delta_{jk}}{\gamma_k}^{(m)}\right)^2$$
$$\leq \left(\frac{\|\gamma^{(m)}\|}{\text{rgap}}\right)^2 \leq \left(\frac{\xi_j}{\text{rgap}}\right)^2 \leq \left(\frac{3\|\gamma^{(1)}\|}{\text{rgap}}\right)^2.$$

Next take the bounds (8.16) and (8.17) on $\|\gamma^{(m)}\|$ and insert them into (8.12) to obtain the desired bound on $\|Z_{kj}^{(m)}\|$, $k \neq j$,

$${\gamma_k}^{(m+1)} \leq \|g_{kj}\| + \frac{\|G_{11}\|}{\text{rgap}_j}{\gamma_k}^{(m)} + f_k \cdot 3\|\gamma^{(1)}\| + \frac{f_j}{\text{rgap}_j}{\gamma_k}^{(m)} \cdot 3\|\gamma^{(1)}\|. \tag{8.18}$$

The right-hand side of (8.18) is linear in ${\gamma_k}^{(m)}$, and we consider the related equality

$$\tau = \mathcal{L}_k(\tau) := \|g_{kj}\| + 3f_k\|\gamma^{(1)}\| + (\|G_{11}\| + \|\gamma^{(1)}\|)\tau/\mathrm{rgap}_j. \tag{8.19}$$

The single positive root τ_{kj} of (8.19) is positive and given in (8.7). Again proceed by induction on m. All coefficients in $\mathcal{L}_k$ are positive. If ${\gamma_k}^{(m)} \le \tau_{kj}$, then, by (8.18),

$${\gamma_k}^{(m+1)} \le \mathcal{L}_k({\gamma_k}^{(m)}) \le \mathcal{L}_k(\tau_{kj}) = \tau_{kj}.$$

Since ${\gamma_k}^{(1)} = \|g_{kj}\| \le \tau_{kj}$, we have, for all $m \ge 1$ and $k \ne j$,

$${\gamma_k}^{(m)} \le \tau_{kj}. \tag{8.20}$$

Equations (8.7) and (8.20) establish (8.6).

Finally, by Lemma 6.1,

$$\|Z^{(m)}\|^2 \le \sum_{k\ne j} \|Z_{kj}^{(m)}\|^2 = \sum_{k\ne j} \left(\frac{\sigma_j {\gamma_k}^{(m)}}{\delta_{jk}}\right)^2 \le \sum_{k\ne j} \left(\frac{\sigma_j \tau_{kj}}{\delta_{jk}}\right)^2. \qquad \square$$

9. Existence of *Z*. The generalized Riccati equation (5.11) that defines Z is equivalent to the $(n-1)$ equations (7.5), with $k \ne j$, and they are quadratic in the unknowns. The technique for proving existence of a solution is known, and we follow Stewart [19]. The sequence $\{Z^m\}$ was defined in (7.6). $\mathbb{R}^{(2n-2)\times 2}$ has finite dimension and thus is compact. The conditions invoked in Theorem 8.1 to provide a bound ξ_j on $\|Z^{(m)}\|$ are sufficient to show that $\{Z^m\}$ is a Cauchy sequence and hence converges to a solution Z of (5.11). These conditions are far from necessary for the existence of Z. Familiarity with section 8 is assumed, including definition of rgap and $\Delta_{\langle j\rangle}$.

THEOREM 9.1. *Assume that ε_d is small enough that the G matrices defined in* (5.3) *and* (5.4) *satisfy the same conditions as in Theorem* 8.1. *Then $\{Z^{(m)}\}$ converges, as $m \longrightarrow \infty$, to a matrix Z that satisfies the generalized Riccati equation* (5.11) *as well as the bounds on $\|Z^{(m)}\|$ given in Theorem* 8.1.

Proof. By Theorem 8.1, $\|Z^{(m)}\| \le \xi_j = 3\|\gamma^{(1)}\|/\mathrm{rgap}$. To prove that $\{Z^m\}$ is a Cauchy sequence take (7.6) for both m and $m-1$ and subtract to obtain

$$\begin{aligned} S_{jk}\left(Z_{kj}^{(m+1)} - Z_{kj}^{(m)}\right) = &-\left(Z_{kj}^{(m)} - Z_{kj}^{(m-1)}\right) F_{11} \\ &+ \sum_{i\ne j} g_{ki}\Phi_i\sigma_i \left(Z_{ij}^{(m)} - Z_{ij}^{(m-1)}\right) \\ &- Z_{kj}^{(m)} F_{12} Z^{(m)} + Z_{kj}^{(m-1)} F_{12} Z^{(m-1)}. \end{aligned} \tag{9.1}$$

As in Theorem 8.1 we must analyze $Z^{(m)}$ at the block level $Z_{kj}^{(m)}$, $k \ne j$, in order to derive the dependence on relative gaps among the $\{\sigma_i\}$ instead of absolute gaps. To this end define $\beta_i^{(m)}$, $i \ne j$, by

$$\|Z_{ij}^{(m)} - Z_{ij}^{(m-1)}\| = \frac{\sigma_j}{\delta_{ji}}\,\beta_i^{(m)}, \quad i \ne j, \tag{9.2}$$

$$\beta^{(m)} = (\beta_1^{(m)}, \ldots, \beta_n^{(m)})^t, \quad \beta_j^{(m)} \text{ omitted.}$$

Compare (9.2) with (8.10), namely,

$$\|Z_{ij}^{(m)}\| = \frac{\sigma_j}{\delta_{ji}}\gamma_i{}^{(m)}, \quad i \ne j.$$

Use Lemma 6.1 to sum the squares of (9.2) to obtain

$$\|Z^{(m)} - Z^{(m-1)}\| \leq \frac{\|\beta^{(m)}\|}{\text{rgap},}$$

and it remains to show that $\|\beta^{(m)}\| \longrightarrow 0$ as $m \longrightarrow \infty$.

We rewrite the two quadratic terms in (9.1):

$$\begin{aligned} Z_{kj}^{(m)} F_{12} Z^{(m)} - Z_{kj}^{(m-1)} F_{12} Z^{(m-1)} &= \left(Z_{kj}^{(m)} - Z_{kj}^{(m-1)}\right) F_{12} Z^{(m)} \\ &\quad + Z_{kj}^{(m-1)} F_{12} \left(Z^{(m)} - Z^{(m-1)}\right). \end{aligned} \tag{9.3}$$

Next we substitute (9.3) into (9.1) and take norms and invoke (9.2). Recall that each F has the form $G\Phi\Sigma$ so that

$$\begin{aligned} \|Z_{kj}^{(m+1)} - Z_{kj}^{(m)}\| \leq \frac{\sigma_j}{\delta_{jk}} \Bigg[& \|Z_{kj}^{(m)} - Z_{kj}^{(m-1)}\| \, \|G_{11}\| \\ & + \sum_{i \neq j} \|g_{ki}\| \sigma_i \frac{\beta_i^{(m)}}{\delta_{ji}} \\ & + \|Z_{kj}^{(m)} - Z_{kj}^{(m-1)}\| \sum_{i \neq j} \|g_{ji}\| \sigma_i \frac{\gamma_i^{(m)}}{\delta_{ji}} \\ & + \|Z_{kj}^{(m-1)}\| \sum_{i \neq j} \|g_{ji}\| \sigma_i \frac{\beta_i^{(m)}}{\delta_{ji}} \Bigg]. \end{aligned} \tag{9.4}$$

We can simplify (9.4) by using f_l from (8.1) and removing the factor σ_j/δ_{jk} to find, after using Cauchy–Schwarz on the sums,

$$\begin{aligned} \beta_k^{(m+1)} \leq \frac{\sigma_j}{\delta_{jk}} \|G_{11}\| \beta_k^{(m)} + f_k \|\beta^{(m)}\| + \frac{\sigma_j}{\delta_{jk}} \beta_k^{(m)} f_j \|\gamma^{(m)}\| \\ + \frac{\sigma_j}{\delta_{jk}} \gamma_k^{(m-1)} f_j \|\beta^{(m)}\|. \end{aligned} \tag{9.5}$$

Next, square (9.5) and sum over $k \neq j$:

$$\begin{aligned} \frac{1}{4} \|\beta^{(m+1)}\|^2 \leq \left(\frac{\|G_{11}\|}{\text{rgap}}\right)^2 \|\beta^{(m)}\|^2 + \left(\sum_{k \neq j} f_k^2\right) \|\beta^{(m)}\|^2 \\ + \left(\frac{f_j \|\gamma^{(m)}\|}{\text{rgap}}\right)^2 \|\beta^{(m)}\|^2 + \left(\frac{f_j \|\beta^{(m)}\|}{\text{rgap}}\right)^2 \|\gamma^{(m-1)}\|^2. \end{aligned} \tag{9.6}$$

So, recall $\|\gamma^{(m)}\| \leq 3\|\gamma^{(1)}\|$ from (8.16) and (8.17) and invoke (8.3):

$$\begin{aligned} \|\beta^{(m+1)}\| \leq 2 \Bigg[\left(\frac{\|G_{11}\|}{\text{rgap}}\right)^2 + \sum_{k \neq j} f_k^2 \\ + 2 \left(\frac{3\|\gamma^{(1)}\| f_j}{\text{rgap}}\right)^2 \Bigg]^{1/2} \|\beta^{(m)}\|. \end{aligned} \tag{9.7}$$

Invoke the hypotheses of Theorem 8.1 to find

$$\|\beta^{(m+1)}\| \le 2\left[\frac{1}{16}+\frac{1}{16}+2\left(\frac{3}{32}\right)^2\right]\|\beta^{(m)}\|$$
$$< \frac{4}{5}\|\beta^{(m)}\|.$$

Thus $\|\beta^{(m)}\| \longrightarrow 0$ as $m \longrightarrow \infty$ and hence $\{Z^{(m)}\}$ is bounded, by Theorem 8.1, and Cauchy and so converges, in $\mathbb{R}^{2(n-1)\times 2}$, to a matrix Z that satisfies (5.11) and the bounds on $\|Z^{(m)}\|$ established in Theorem 8.1. □

10. The combined bounds. Recall from section 5 that Z is the solution of the generalized Riccati equation (5.11) and determines the similarity N that diagonalizes the perturbed double matrix $C+F$ in (5.3). All that remains is to insert the bounds on $\|Z\|$ from section 8 into the earlier results of section 6. We take the opportunity to recapitulate earlier definitions so that this section is somewhat independent of the preceding analysis.

An $n \times n$ invertible matrix K has Ω-SVD $K = U\Sigma V^t$, and our interest is the Ω-SVD of a scaled matrix $D_l K D_r$.

For the scaling matrices D_l and D_r, the perturbation parameter (see (3.2)) is ε_d, d for diagonal scaling. The bounds in Theorem 10.1 contain no explicit factors of n, but when they are applied to a bidiagonal matrix, then $\varepsilon_d = (1+\varepsilon)^{2n-1} - 1$, with ε the roundoff unit (see (3.2)),

$$\bar{\varepsilon}_d = \varepsilon_d(2+\varepsilon_d) \quad \text{(bounds } \|D_l^2 - I\| \text{ etc. from Lemma 6.3)},$$
$$\text{rgap} = \text{rgap}_j := \min_{k\neq j} \frac{\delta_{jk}}{\sigma_j},$$
$$\delta_{jk} = \left\{ \begin{array}{ll} |\sigma_j - \sigma_k| & \text{if } \omega_k = \omega_j, \\ \frac{\sigma_j^2+\sigma_k^2}{\sigma_j+\sigma_k} & \text{if } \omega_k \neq \omega_j. \end{array} \right\} \text{from (7.4)}.$$

Note that

$$|\sigma_j - \sigma_k| < \frac{\sigma_j^2+\sigma_k^2}{\sigma_j+\sigma_k} < \sigma_j + \sigma_k.$$

Next we introduce new quantities needed to express our bounds. The conditions in Theorems 8.1 and 9.1 (they are the same) are constraints on the G matrices of Lemma 5.1. The 2×2 blocks of these matrices were bounded in (5.6), which we repeat here, and then use immediately. With $\|D_l - I\| \le \varepsilon_d$ and $\|D_r - I\| \le \varepsilon_d$,

$$\|g_{ik}\| \le \bar{\varepsilon}_d \max\{1, \|\boldsymbol{v}_i\|\,\|\boldsymbol{v}_k\|\} = \bar{\varepsilon}_d \|\boldsymbol{v}_i\|\,\|\boldsymbol{v}_k\|, \tag{10.1}$$

since

$$\|\boldsymbol{v}_i\| \ge 1, \quad i = 1, \ldots, n.$$

Recall that the subscript $\langle j\rangle$ denotes that the jth item is omitted. In section 8 we

introduced f_k, $k = 1, \ldots, n$, which we now bound:

$$f_k = \left(\sum_{i \neq j} \left(\|g_{ki}\| \sigma_i / \delta_{ji} \right)^2 \right)^{1/2}$$

$$\leq \bar{\varepsilon}_d \|\boldsymbol{v}_k\| \|\boldsymbol{m}_{\langle j \rangle}\|, \tag{10.2}$$

with

$$\boldsymbol{m}_{\langle j \rangle} := (m_{\langle j \rangle}(1), \ldots, m_{\langle j \rangle}(n))^t, \quad m_{\langle j \rangle}(j) \quad \text{omitted},$$
$$m_{\langle j \rangle}(i) := \|\boldsymbol{v}_i\| \sigma_i / \delta_{ji}.$$

Now we can bound the left-hand sides in conditions (8.2)–(8.4) of Theorem 8.1. Condition (8.2):

$$\|G_{11}\| = \|g_{jj}\| \leq \bar{\varepsilon}_d \|\boldsymbol{v}_j\|^2. \tag{10.3}$$

Condition (8.3):

$$\left(\sum_{i \neq j} f_i^2 \right)^{1/2} \leq \bar{\varepsilon}_d \|\boldsymbol{m}_{\langle j \rangle}\| \, \|V_{\langle j \rangle}\|_F. \tag{10.4}$$

Condition (8.4):

$$f_j \|\gamma^{(1)}\| \leq (\bar{\varepsilon}_d \|\boldsymbol{v}_j\| \, \|\boldsymbol{m}_{\langle j \rangle}\|)(\bar{\varepsilon}_d \|\boldsymbol{v}_j\| \, \|V_{\langle j \rangle}\|_F)$$

$$= (\bar{\varepsilon}_d \|\boldsymbol{v}_j\|)^2 \|\boldsymbol{m}_{\langle j \rangle}\| \, \|V_{\langle j \rangle}\|_F, \tag{10.5}$$

since

$$\|G_{21}\| \leq \|\gamma^{(1)}\| := \left[\sum_{i \neq j} \|g_{ij}\|^2 \right]^{1/2}.$$

The dominant term in Z is $Z^{(1)}$, and

$$\|Z^{(1)}\| = \left(\sum_{k \neq j} \left(\|g_{kj}\| \sigma_j / \delta_{jk} \right)^2 \right)^{1/2} \leq \bar{\varepsilon}_d \|\boldsymbol{m}_j\| \, \|\boldsymbol{v}_j\|, \tag{10.6}$$

with

$$\boldsymbol{m}_j = (m_j(1), \ldots, m_j(n))^t, \quad m_j(j) \quad \text{omitted}, \tag{10.7}$$
$$m_j(i) = \|\boldsymbol{v}_i\| \sigma_j / \delta_{ji}.$$

Note the important difference between $\boldsymbol{m}_j$ and $\boldsymbol{m}_{\langle j \rangle}$ in (10.2) and (10.7). It is easy to see that $\|\boldsymbol{m}_j\| \leq \|V_{\langle j \rangle}\|_F / \mathrm{rgap}_j$, but the example in section 11 has $\|\boldsymbol{m}_j\| \ll \|V_{\langle j \rangle}\|_F / \mathrm{rgap}_j$, and this discrepancy led us to the detailed analysis that yielded (10.6). The term $\|\boldsymbol{m}_j\|$ is the reward for the detailed analysis begun in section 7. The important bounds in Theorem 10.1 are (10.14) and (10.15).

One more definition simplifies the statement of Theorem 10.1, a messy quantity close to 1:

$$\Gamma_j := \frac{1 + 3\bar{\varepsilon}_d \|\boldsymbol{m}_{\langle j\rangle}\| \, \|V_{\langle j\rangle}\|_F}{1 - \bar{\varepsilon}_d \|\boldsymbol{v}_j\|^2 (1 + 3\bar{\varepsilon}_d \|\boldsymbol{m}_{\langle j\rangle}\| \, \|V_{\langle j\rangle}\|_F)/\mathrm{rgap}_j}. \tag{10.8}$$

THEOREM 10.1. *Consider a signature matrix* $\Omega = \mathrm{diag}(\omega_1, \ldots, \omega_n)$, $\omega_i = \pm 1$ *and an* $n \times n$ *invertible matrix* K *with* Ω*-SVD*

$$\begin{aligned} K = U\Sigma V^t, \quad U^t = U^{-1}, \quad V^t \Omega V = \Omega, \\ U = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n], \quad V = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n], \\ \Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n), \quad \sigma_i > 0, \end{aligned}$$

and with all $\{\omega_i \sigma_i^2\}$ *distinct from each other. Let* D_l *and* D_r *be diagonal scaling matrices satisfying*

$$\|D_l^2 - I\| \le \bar{\varepsilon}_d, \;\; \|D_r^2 - I\| \le \bar{\varepsilon}_d, \;\; \|D_l^{-2} - I\| \le \bar{\varepsilon}_d, \;\; \|D_r^{-2} - I\| \le \bar{\varepsilon}_d.$$

If the perturbation parameter $\bar{\varepsilon}_d$ *is small enough that the following conditions invoking* (10.3)–(10.5)*,*

$$8\bar{\varepsilon}_d \|\boldsymbol{v}_j\|^2 \le \mathrm{rgap}_j, \tag{10.9}$$

$$4\bar{\varepsilon}_d \|\boldsymbol{m}_{\langle j\rangle}\| \, \|V_{\langle j\rangle}\|_F \le 1, \tag{10.10}$$

hold, then

$$\|Z\| \le \bar{\varepsilon}_d \|\boldsymbol{v}_j\| \, \|\boldsymbol{m}_j\| \Gamma_j \le 2\bar{\varepsilon}_d \|\boldsymbol{v}_j\| \, \|\boldsymbol{m}_j\|, \tag{10.11}$$

and there is an Ω *singular triple* $(\tilde{\sigma}_j, \tilde{\boldsymbol{u}}_j, \Omega\tilde{\boldsymbol{v}}_j)$ *of* $D_l K D_r$ *that satisfies*

$$\frac{|\tilde{\sigma}_j^2 - \sigma_j^2|}{\sigma_j^2} \le \frac{\bar{\varepsilon}_d(\|\boldsymbol{v}_j\|^2 + 1) + \beta_2}{1 - \bar{\varepsilon}_d(\|\boldsymbol{v}_j\|^2 + 1) - \beta_2}, \tag{10.12}$$

$$\beta_2 \le 2(\bar{\varepsilon}_d \|\boldsymbol{v}_j\|)^2 \left[(\|V_{\langle j\rangle}\|_F + 2\|\boldsymbol{m}_j\|)^2 + 4\|\boldsymbol{m}_j\| \, \|\boldsymbol{m}_{\langle j\rangle}\| \right] \tag{10.13}$$

so that

$$\frac{|\tilde{\sigma}_j^2 - \sigma_j^2|}{\sigma_j^2} \le \bar{\varepsilon}_d(\|\boldsymbol{v}_j\|^2 + 1) + O\left((\bar{\varepsilon}_d \|\boldsymbol{v}_j\| \, \|V_{\langle j\rangle}\|_F)^2 \right), \tag{10.14}$$

$$\begin{aligned} |\sin \angle(\tilde{\boldsymbol{u}}_j, \boldsymbol{u}_j)| &\le \frac{\|Z\| + \varepsilon_d}{(1 - \varepsilon_d)^2} \\ &\le \frac{\bar{\varepsilon}_d(2\|\boldsymbol{v}_j\|\|\boldsymbol{m}_j\| + \frac{1}{2})}{1 - \bar{\varepsilon}_d}. \end{aligned} \tag{10.15}$$

If, in addition,

$$\bar{\beta}_j := \left(\|\boldsymbol{v}_j\| - \sqrt{\|\boldsymbol{v}_j\|^4 - 1} \|V_{\langle j\rangle}\| \, \|Z\| \right)^{-1} > 0, \tag{10.16}$$

then

$$|\sin\angle(\Omega\tilde{\boldsymbol{v}}_j, \Omega\boldsymbol{v}_j)| \le \frac{(\boldsymbol{\kappa}_j + \varepsilon_d)(1 + \varepsilon_d\bar{\beta}_j)}{1 - \varepsilon_d},$$

$$\boldsymbol{\kappa}_j := \|V_{\langle j\rangle}\|\,\|Z\|\bar{\beta}_j. \tag{10.17}$$

If $\bar{\varepsilon}_d$ is small enough that $\bar{\beta}_j \le 2/\|\boldsymbol{v}_j\|$, then

$$|\sin\angle(\Omega\tilde{\boldsymbol{v}}_j, \Omega\boldsymbol{v}_j)| \le \frac{\bar{\varepsilon}_d(4\|V_{\langle j\rangle}\|\,\|\boldsymbol{m}_j\| + \frac{1}{2})}{(1 - \bar{\varepsilon}_d)^{3/2}}. \tag{10.18}$$

Proof. Conditions (10.9) and (10.10) and their product guarantee conditions (8.2)–(8.4) in Theorem 8.1, which in turn guarantee the existence of $Z = \lim_{m\to\infty} Z^{(m)}$ by Theorem 9.1, and hence the validity of the bounds in Theorems 8.1 on $\|Z\|$ and $\|Z_{kj}\|$.

First we derive a bound on $\|Z\|$. From Theorems 8.1 and 9.1,

$$\|Z\|^2 \le \sum_{k\ne j}\left(\frac{\sigma_j\tau_{kj}}{\delta_{jk}}\right)^2, \tag{10.19}$$

$$\tau_{kj} = \frac{\|g_{kj}\| + 3f_k\|\gamma^{(1)}\|}{1 - (\|G_{11}\| + 3f_j\|\gamma^{(1)}\|)/\mathrm{rgap}_j}. \tag{10.20}$$

Use (10.3)–(10.5) in (10.20) to obtain

$$\tau_{kj} \le \frac{\bar{\varepsilon}_d\|\boldsymbol{v}_j\|\,\|\boldsymbol{v}_k\|(1 + 3\bar{\varepsilon}_d\|\boldsymbol{m}_{\langle j\rangle}\|\,\|V_{\langle j\rangle}\|_F)}{1 - \bar{\varepsilon}_d\|\boldsymbol{v}_j\|^2(1 + 3\bar{\varepsilon}_d\|\boldsymbol{m}_{\langle j\rangle}\|\,\|V_{\langle j\rangle}\|_F)/\mathrm{rgap}_j}$$

$$= \bar{\varepsilon}_d\|\boldsymbol{v}_j\|\,\|\boldsymbol{v}_k\|\Gamma_j, \tag{10.21}$$

with Γ_j defined in (10.8).

Insert (10.21) into (10.19) to find

$$\|Z\| \le \bar{\varepsilon}_d\|\boldsymbol{v}_j\|\,\|\boldsymbol{m}_j\|\Gamma_j,$$

which is (10.11).

Conditions (10.9) and (10.10) imply that Γ_j, generally close to 1, is bounded by 2. From (10.8),

$$\Gamma_j \le \frac{1 + 3(1/4)}{1 - (1/8)(3/4)} = \frac{56}{29} < 2.$$

This completes (10.11).

Next we bound the change in angles. Lemma 6.4 in section 6 gives

$$\begin{aligned}|\sin\angle(\tilde{\boldsymbol{u}}_j, \boldsymbol{u}_j)| &\le \frac{\|Z\| + \varepsilon_d}{(1 - \varepsilon_d)^2}\\ &\le \frac{2\bar{\varepsilon}_d\|\boldsymbol{v}_j\|\,\|\boldsymbol{m}_j\| + \varepsilon_d}{(1 - \varepsilon_d)^2}\\ &\le \frac{\bar{\varepsilon}_d(2\|\boldsymbol{v}_j\|\,\|\boldsymbol{m}_j\| + \frac{1}{2})}{1 - \bar{\varepsilon}_d},\end{aligned}$$

which is (10.15).

Lemma 6.5 in section 6 gives

$$|\sin\angle(\Omega\tilde{\boldsymbol{v}}_j, \Omega\boldsymbol{v}_j)| \le [\boldsymbol{\kappa}_j + \varepsilon_d(1+\boldsymbol{\kappa}_j)](1+\varepsilon_d\bar{\beta}_j),$$

$$\boldsymbol{\kappa}_j := \|V_{\langle j\rangle}\|\,\|\boldsymbol{z}_v\|\bar{\beta}_j,$$

with

$$\bar{\beta}_j := \left(\|\boldsymbol{v}_j\| - \sqrt{\|\boldsymbol{v}_j\|^2 - 1}\|V_{\langle j\rangle}\|\,\|\boldsymbol{z}_v\|\right)^{-1}.$$

Use $\|\boldsymbol{z}_v\| \le \|Z\|$ and observe that $[\boldsymbol{\kappa}_j + \varepsilon_d(1+\boldsymbol{\kappa}_j)] \le (\boldsymbol{\kappa}_j + \varepsilon_d)/(1-\varepsilon_d)$ to get (10.17). Finally if $\bar{\beta}_j \le 2/\|\boldsymbol{v}_j\|$, then we have

$$\boldsymbol{\kappa}_j \le \|V_{\langle j\rangle}\|2\bar{\varepsilon}_d\|\boldsymbol{v}_j\|\,\|\boldsymbol{m}_j\|(2/\|\boldsymbol{v}_j\|).$$

Use

$$1 + \varepsilon_d\bar{\beta}_j \le 1 + \frac{2\bar{\varepsilon}_d}{\|\boldsymbol{v}_j\|} \le 1 + \bar{\varepsilon}_d \le (1-\bar{\varepsilon}_d)^{-1} \tag{10.22}$$

to establish (10.18).

Now consider $\tilde{\sigma}_j$ and Lemma 6.3. It is only necessary to bound β_2:

$$\beta_2 = (\bar{\varepsilon}_d\Upsilon_j + \|Z\|_F)^2 + 2|\boldsymbol{\kappa}_{\langle j\rangle}|, \tag{10.23}$$

$$\begin{aligned}
|\boldsymbol{\kappa}_{\langle j\rangle}| &= \sigma_j^{-1}\left|\sum_{i\ne j}\omega_i\sigma_i\det[Z_{ij}]\right|, \quad \text{from Lemma 6.1,}\\
&\le \sigma_j^{-1}\sum_{i\ne j}\sigma_i\|Z_{ij}\|^2\\
&\le \sigma_j^{-1}\sum_{i\ne j}\sigma_i\left(2\bar{\varepsilon}_d\|\boldsymbol{v}_j\|\,\|\boldsymbol{v}_i\|\frac{\sigma_j}{\delta_{ji}}\right)^2, \quad \text{by (10.21),}\\
&\le (2\bar{\varepsilon}_d\|\boldsymbol{v}_j\|)^2\sum_{i\ne j}\left(\frac{\|\boldsymbol{v}_i\|\sigma_i}{\delta_{ji}}\right)\left(\frac{\|\boldsymbol{v}_i\|\sigma_j}{\delta_{ji}}\right)\\
&\le (2\bar{\varepsilon}_d\|\boldsymbol{v}_j\|)^2\,\|\boldsymbol{m}_j\|\,\|\boldsymbol{m}_{\langle j\rangle}\|, \quad \text{by Cauchy–Schwarz,} \qquad (10.24)
\end{aligned}$$

$$\Upsilon_j^2 := (n-1) + \|\boldsymbol{v}_j\|^2\,\|V_{\langle j\rangle}\|_F^2 \le 2(\|\boldsymbol{v}_j\|\,\|V_{\langle j\rangle}\|_F)^2, \tag{10.25}$$

$$\|Z\|_F^2 = \|\boldsymbol{z}_u\|^2 + \|\boldsymbol{z}_v\|^2 \le 2\max\{\|\boldsymbol{z}_u\|^2, \|\boldsymbol{z}_v\|^2\} = 2\|Z\|^2. \tag{10.26}$$

Hence, inserting (10.11), (10.24), (10.25), and (10.26) into (10.23),

$$\beta_2 \le 2(\bar{\varepsilon}_d\|\boldsymbol{v}_j\|)^2\left[(\|V_{\langle j\rangle}\|_F + 2\|\boldsymbol{m}_j\|)^2 + 4\|\boldsymbol{m}_j\|\,\|\boldsymbol{m}_{\langle j\rangle}\|\right],$$

which gives (10.13) and, by Lemma 6.1, (10.14). ∎

Note that $\|\boldsymbol{m}_{\langle j\rangle}\|$ and $\|\boldsymbol{m}_j\|$ involve all the $\|\boldsymbol{v}_i\|$. When $\|\boldsymbol{v}_j\|$ is close to 1, but $\|V_{\langle j\rangle}\|_F$ is huge, there is a natural concern that $\|\boldsymbol{m}_j\|$ might also be large, thus degrading the bound on $|\sin\angle(\tilde{\boldsymbol{u}}_j, \boldsymbol{u}_j)|$. However, the relative gaps δ_{jk}/σ_j can also be huge (take $\sigma_j = 10^{-8}$, $\sigma_k = 1$) and can neutralize large values of $\|\boldsymbol{v}_k\|$.

The quantity $\|\boldsymbol{m}_{\langle j\rangle}\|\,\|V_{\langle j\rangle}\|_F$ occurs in the second order ($O(\bar{\varepsilon}_d^2)$) terms. Now $\|\boldsymbol{m}_{\langle j\rangle}\| \le \|V_{\langle j\rangle}\|_F / \min_k(\delta_{jk}/\sigma_k)$, and consequently the stability of the Ω singular triple $(\sigma_j, \boldsymbol{u}_j, \Omega\boldsymbol{v}_j)$ depends on $\|V\|_F$ and the *relative* separations among the $\{\sigma_i\}$. Finally, recall from Lemma 2.1 that $\|V_{\langle j\rangle}\|_F < \|V\|_F \le \sqrt{\mathrm{cond}_F(K)}$. Nevertheless for $\Omega = I$, V is orthogonal whatever the value of $\mathrm{cond}_F(K)$, and, in general, the coupling between them is weak.

When $\Omega = I$, then all $\|\boldsymbol{v}_i\| = 1$ and

$$\|\boldsymbol{m}_j\| = \left[\sum_{k\neq j}\left(\frac{\sigma_j}{\delta_{jk}}\right)^2\right]^{1/2}, \quad \|\boldsymbol{m}_{\langle j\rangle}\| = \left[\sum_{k\neq j}\left(\frac{\sigma_k}{\delta_{jk}}\right)^2\right]^{1/2}.$$

The conclusions, to the first order, are

$$\frac{|\tilde{\sigma}_j^2 - \sigma_j^2|}{\sigma_j^2} \le 4\bar{\varepsilon}_d,$$

$$|\sin\angle(\tilde{\boldsymbol{u}}_j, \boldsymbol{u}_j)| \le \left(2\|\boldsymbol{m}_j\| + \frac{1}{2}\right)\bar{\varepsilon}_d,$$

$$|\sin\angle(\tilde{\boldsymbol{v}}_j, \boldsymbol{v}_j)| \le \left(4\|\boldsymbol{m}_j\| + \frac{1}{2}\right)\bar{\varepsilon}_d, \quad \text{since } \|V_{\langle j\rangle}\| = 1.$$

These results are worse than the Demmel–Kahan results in [3] by a factor of about 4.

The bounds in Theorem 10.1 are quite realistic and suggest the following definitions for condition numbers relating to the effect of small relative changes in L's entries on an eigenpair $(\lambda_j, \boldsymbol{u}_j)$ of $L\Omega L^t$:

$$\mathrm{relcond}(\lambda_j) := \|\boldsymbol{v}_j\|^2 + 1, \tag{10.27}$$

$$\mathrm{relcond}(\boldsymbol{u}_j) := 2\|\boldsymbol{v}_j\|\,\|\boldsymbol{m}_j\| + \frac{1}{2}, \tag{10.28}$$

with $\|\boldsymbol{m}_j\|$ given in (10.7).

We emphasize that only the eigenvalues with the same sign as λ_j bring true relative gaps to $\|\boldsymbol{m}_j\|$. When $\omega_i \neq \omega_j$, then

$$\frac{\sigma_i}{\delta_{ji}} + \frac{\sigma_j}{\delta_{ji}} \le 2,$$

and the ith entries of $\boldsymbol{m}_j$ and $\boldsymbol{m}_{\langle j\rangle}$ sum to $2\|\boldsymbol{v}_i\|$.

11. An example. In another article [14] we presented, in detail, a 4×4 symmetric matrix $T = T(\eta)$ that depends on a small parameter η. T is indefinite but permits triangular factorization $T = L\Omega L^t$ with huge element growth, like $1/\eta$, in the multipliers. This ill-conditioned L has Ω-SVD $L = U\Sigma V^t$, with $\|V\|_F^2 = 1/\eta$. The two small σ's are close, 1.15η and 1.65η, while the other σ's are almost 1 but have differing ω values. The large singular values are extremely sensitive (condition number $1/\eta$), but the two small σ's are robust and the associated singular vectors are also robust, their $\|\boldsymbol{m}\|$ values showing the neutralizing of each large $\|\boldsymbol{v}\|$ by an equally

large relative gap:

$$T = T(\eta) := \begin{bmatrix} \eta & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -2\eta & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 3\eta & \eta \\ 0 & 0 & \eta & 2\eta \end{bmatrix},$$

$$T = L\Omega L^t = U(\Sigma^2\Omega)U^t,$$

$$\Omega = \operatorname{diag}(1,-1,1,1), \quad L = U\Sigma V^t.$$

We present only the leading terms in the quantities shown below. The eigenvalues $\omega\sigma^2$ are not presented in monotonic order because of the constraint that $V^t\Omega V = \Omega$:

$$\Lambda = \Sigma^2\Omega = \operatorname{diag}\left(\frac{4-\sqrt{2}}{2}\eta, -1, \frac{4+\sqrt{2}}{2}\eta, +1\right),$$

$$\Sigma = \operatorname{diag}(\sqrt{\eta}\,\mu_-, 1, \sqrt{\eta}\,\mu_+, 1), \quad \mu_-^2 := \frac{4-\sqrt{2}}{2}, \quad \mu_+^2 := \frac{4+\sqrt{2}}{2},$$

$$V = \begin{bmatrix} 1 & -\frac{1}{2} & 1 & \frac{1}{2} \\ 1 & -\frac{1+2\eta}{2} & 1 & \frac{1-2\eta}{2} \\ -1 & \eta\left(1-\frac{9}{4}\eta\right) & -1 & \eta\left(1+\frac{9}{4}\eta\right) \\ \frac{\sqrt{14}}{4-\sqrt{2}} & -\frac{\sqrt{7}}{4}\eta^2 & -\frac{\sqrt{14}}{4+\sqrt{2}} & \frac{\sqrt{7}}{4}\eta^2 \end{bmatrix} \begin{bmatrix} \frac{1}{2}\mu_- & 0 & 0 & \\ 0 & \frac{1}{\sqrt{\eta}} & 0 & 0 \\ 0 & 0 & \frac{1}{2}\mu_+ & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{\eta}}. \end{bmatrix},$$

$$\|\boldsymbol{v}_i\|^2 = \; 2-\tfrac{\sqrt{2}}{4}, \quad \tfrac{1}{2\eta}, \quad 2+\tfrac{\sqrt{2}}{4}, \quad \tfrac{1}{2\eta}.$$

For $\sigma_1 = \sqrt{\eta}\,\mu_- = 1.14\sqrt{\eta}$,

$$\operatorname{rgap}_1 = \frac{\mu_+ - \mu_-}{\mu_-} = \frac{4}{9},$$

$$\|V_{<1>}\|_F = \frac{1}{\sqrt{\eta}}\left[1+\eta\left(2+\frac{\sqrt{2}}{4}\right)\right]^{1/2},$$

$$\Delta_{<1>} = \left(\bullet, \frac{1+\sqrt{\eta}\mu_-}{(1+\eta\mu_-^2)^{1/2}}, \sqrt{\eta}(\mu_+ - \mu_-), 1-\sqrt{\eta}\mu_-\right).$$

The second entry in $\Delta_{<1>}$ is a quotient, not a difference, because $\omega_1 \neq \omega_2$, but the denominator is $1 + O(\eta)$ and contributes only higher order effects. Observe the

neutralizing effect in $\|\boldsymbol{m}_1\|$ below:

$$\begin{aligned}\|\boldsymbol{m}_1\| &= \left[\sum_{i\neq 1}\|\boldsymbol{v}_i\|^2\left(\sigma_1/\delta_{1i}\right)^2\right]^{1/2}\\ &= \left[\frac{1}{2\eta}\left(\frac{\sqrt{\eta}\mu_-}{1+\sqrt{\eta}\mu_-}\right)^2+\left(2+\frac{\sqrt{2}}{4}\right)\left(\frac{\sqrt{\eta}\mu_-}{\sqrt{\eta}(\mu_+-\mu_-)}\right)^2\right.\\ &\quad\left.+\frac{1}{2\eta}\left(\frac{\sqrt{\eta}\mu_-}{1-\sqrt{\eta}\mu_-}\right)^2\right]^{1/2}\\ &= \left[\frac{1}{2}\mu_-^2+\left(2+\frac{\sqrt{2}}{4}\right)\left(\frac{\mu_-}{\mu_+-\mu_-}\right)^2+\frac{1}{2}\mu_-^2\right]^{1/2}\\ &= (\mu_-)\left[1+(8+\sqrt{2})\right]^{1/2}\approx\frac{17}{4}.\end{aligned}$$

On the other hand,

$$\|\boldsymbol{m}_{<1>}\| = \left[\sum_{i\neq 1}\|\boldsymbol{v}_i\|^2\left(\sigma_i/\delta_{1i}\right)^2\right]^{1/2} = \frac{1}{\sqrt{\eta}}\left[1+\eta\left(2+\frac{\sqrt{2}}{4}\right)\left(\frac{\mu_+}{\mu_+-\mu_-}\right)^2\right]^{1/2}.$$

Observe that $\|\boldsymbol{m}_1\|\leq\|V_{<1>}\|_F/\mathrm{rgap}_1\approx 9/(4\sqrt{\eta})$ is true but excessively pessimistic.

The conditions of Theorem 10.1 for $(\sigma_1,\boldsymbol{u}_1,\Omega\boldsymbol{v}_1)$ are

$$\begin{aligned}8\bar{\varepsilon}_d(1.65)\leq\mathrm{rgap}_1=\frac{4}{9},\\ 4\bar{\varepsilon}_d\eta^{-1/2}\eta^{-1/2}\leq 1,\end{aligned}$$

and the conclusions are

$$\begin{aligned}\frac{|\tilde{\sigma}_1^2-\sigma_1^2|}{\sigma_1^2}&\leq(\|\boldsymbol{v}_1\|^2+1)\bar{\varepsilon}_d=2.65\bar{\varepsilon}_d,\\ |\sin\angle(\tilde{\boldsymbol{u}}_1,\boldsymbol{u}_1)|&\leq(2\|\boldsymbol{m}_1\|\|\boldsymbol{v}_1\|+1/2)\bar{\varepsilon}_d=11.5\bar{\varepsilon}_d,\\ |\sin\angle(\Omega\tilde{\boldsymbol{v}}_1,\Omega\boldsymbol{v}_1)|&\leq(4\|\boldsymbol{m}_1\|\|V_{<1>}\|+1/2)\varepsilon_d=17(\bar{\varepsilon}_d/\sqrt{\eta}).\end{aligned}$$

For $\sigma_2=1$, with $\omega_2=-1$,

$$\begin{aligned}\mathrm{rgap}_2&=\frac{\sigma_2+\sigma_4}{\sigma_2^2+\sigma_4^2}=1,\\ \|V_{<2>}\|_F&=\frac{1}{\sqrt{2\eta}},\\ \Delta_{<2>}&=\left(\frac{1+\sqrt{\eta}\mu_-}{1+\eta\mu_-^2},\bullet,\frac{1+\sqrt{\eta}\mu_+}{1+\eta\mu_+^2},\frac{1+1}{1+1}\right),\end{aligned}$$

$$\|\boldsymbol{m}_2\| = \left[\left(2-\frac{\sqrt{2}}{4}\right)\left(\frac{1}{1+\sqrt{\eta}\mu_-}\right)^2 + \left(2+\frac{\sqrt{2}}{4}\right)\left(\frac{1}{1+\sqrt{\eta}\mu_+}\right)^2 + \frac{1}{2\eta}(1)^2\right]^{1/2}$$
$$\approx \frac{1}{\sqrt{2\eta}}\,(1+8\eta)^{1/2},$$

$$\|\boldsymbol{m}_{<2>}\| = \left[\left(2-\frac{\sqrt{2}}{4}\right)\left(\frac{\sqrt{\eta}\mu_-}{1+\sqrt{\eta}\mu_-}\right)^2 + \left(2+\frac{\sqrt{2}}{4}\right)\left(\frac{\sqrt{\eta}\mu_+}{1+\sqrt{\eta}\mu_+}\right)^2 + \frac{1}{2\eta}(1)^2\right]^{1/2}$$
$$\approx \frac{1}{\sqrt{2\eta}}\left[1+O(\eta^2)\right]^{1/2}.$$

The conditions of Theorem 10.1 for $(\sigma_2, \boldsymbol{u}_2, \Omega\boldsymbol{v}_2)$ are

$$8\bar{\varepsilon}_d\left(\frac{1}{2\eta}\right) \le \mathrm{rgap}_2 = 1,$$
$$4\bar{\varepsilon}_d\left(\frac{1}{2\eta}\right) \le 1,$$

and the conclusions are

$$\frac{|\tilde{\sigma}_2^2 - \sigma_2^2|}{\sigma_2^2} \le (1/(2\eta)+1)\bar{\varepsilon}_d,$$
$$|\sin\angle(\tilde{\boldsymbol{u}}_2, \boldsymbol{u}_2)| \le (1/\eta + 1/2)\bar{\varepsilon}_d,$$
$$|\sin\angle(\Omega\tilde{\boldsymbol{v}}_2, \Omega\boldsymbol{v}_2)| \le (2/\eta + 1/2)\bar{\varepsilon}_d.$$

The results for σ_3 (respectively, σ_4) are similar to those for σ_1 (respectively, σ_2). Thus if $\eta = \sqrt{\varepsilon}$, then σ_2 and σ_4 and their vectors are only defined to half working precision.

REFERENCES

[1] A. W. Bojanczyk, R. Onn, and A. O. Steihardt, *Existence of the hyperbolic singular value decomposition*, Linear Algebra Appl., 185 (1993), pp. 21–30.

[2] P. Deift, J. Demmel, L.-C. Li, and C. Tomei, *The bidiagonal singular value decomposition and Hamiltonian mechanics*, SIAM. J. Numer. Anal., 28 (1991), pp. 1463–1516.

[3] J. Demmel and W. Kahan, *Accurate singular values of bidiagonal matrices*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 873–912.

[4] I. S. Dhillon, *A New $O(n^2)$ Algorithm for the Symmetric Tridiagonal Eigenvalue/Eigenvector Problem*, Ph.D. thesis (UCB//CSD-97-971), Computer Science Division, University of California, Berkeley, CA, 1997.

[5] I. S. Dhillon and B. N. Parlett, *Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices*, Linear Algebra Appl., 387 (2004), pp. 1–28.

[6] I. S. Dhillon and B. N. Parlett, *Orthogonal eigenvectors and relative gaps*, SIAM J. Matrix Anal. Appl. 25 (2004) pp. 858-899.

[7] S. Eisenstat and I. C. F. Ipsen, *Relative perturbation bounds for eigenspaces and singular vector subspaces*, in Proceedings of the Fifth SIAM Conference on Applied Linear Algebra, Proceedings in Applied Mathematics 72, J. Lewis, ed., SIAM, Philadelphia, 1994, pp. 62–65.

[8] S. C. Eisenstat and I. C. F. Ipsen, *Relative perturbation techniques for singular value problems*, SIAM J. Numer. Anal., 32 (1995) pp. 1972-1988.

[9] G. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM J. Numer. Anal., 2 (1965), pp. 205–224.
[10] R.-C. Li, *Relative perturbation theory.* III. *More bounds on eigenvalue variation*, Linear Algebra Appl., 266 (1997), pp. 337–345.
[11] R.-C. Li, *Relative perturbation theory:* I. *Eigenvalue and singular value variations*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 956–982.
[12] R.-C. Li, *Relative perturbation theory:* II. *Eigenspace and singular subspace variations*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 471–492.
[13] R. Onn, A. O. Steihardt, and A. W. Bojanczyk, *The hyperbolic singular value decomposition and applications*, IEEE Trans. Signal Process., 39 (1991), pp. 1575–1588.
[14] Beresford N. Parlett, *Perturbations of eigenpairs of factored symmetric tridiagonal matrices*, Found. Comput. Math., 3 (2003) pp. 207-223.
[15] I. Slapničar, *Highly accurate symmetric eigenvalue decomposition and hyperbolic SVD*, Linear Algebra Appl., 358 (2003), pp. 387–424.
[16] I. Slapničar and N. Truhar, *Relative perturbation theory for hyperbolic eigenvalue problem*, Linear Algebra Appl., 309 (2000), pp. 57–72.
[17] I. Slapničar and N. Truhar, *Relative perturbation theory for hyperbolic singular value problem*, Linear Algebra Appl., 358 (2003), pp. 367–386.
[18] I. Slapničar and K. Veselic, *A bound for the condition of a hyperbolic eigenvector matrix*, Linear Algebra Appl., 290 (1999), pp. 247–255.
[19] G. W. Stewart, *Matrix Algorithms Volume* II*: Eigensystems*, SIAM, Philadelphia, 2001.
[20] K. Veselić, *Perturbation theory for the eigenvalues of factorised symmetric matrices*, Linear Algebra Appl., 309 (2000), pp. 85–102.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 4, pp. 1058–1082

A NUMERICAL METHOD FOR COMPUTING AN SVD-LIKE DECOMPOSITION*

HONGGUO XU†

Abstract. We present a numerical method for computing the SVD-like decomposition $B = QDS^{-1}$, where Q is orthogonal, S is symplectic, and D is a permuted diagonal matrix. The method can be applied directly to compute the canonical form of the Hamiltonian matrices of the form JB^TB, where $J = \left[\begin{smallmatrix} 0 & I \\ -I & 0 \end{smallmatrix}\right]$. It can also be applied to solve the related application problems such as the gyroscopic systems and linear Hamiltonian systems. Error analysis and numerical examples show that the eigenvalues of JB^TB computed by this method are more accurate than those computed by the methods working on the explicit product JB^TB or BJB^T.

Key words. skew-symmetric matrix, Hamiltonian matrix, symplectic matrix, orthogonal symplectic matrix, eigenvalue problem, SVD, SVD-like decomposition, Schur form, Jordan canonical form, QR algorithm, Jacobi algorithm

AMS subject classification. 65F15

DOI. 10.1137/S0895479802410529

1. Introduction.

It is shown in [18] that every real matrix $B \in \mathbb{R}^{n\times 2m}$ has an SVD-like decomposition

$$Q^TBS = \begin{array}{c} \\ p \\ q \\ p \\ n-2p-q \end{array} \begin{array}{c} \begin{array}{cccccc} p & q & m-p-q & p & q & m-p-q \end{array} \\ \begin{pmatrix} \Sigma & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Sigma & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}, \tag{1.1}$$

where matrix Q is real orthogonal, S is real symplectic, and Σ is positive diagonal.

DEFINITION 1.1. *Let* $J = \left[\begin{smallmatrix} 0 & I_m \\ -I_m & 0 \end{smallmatrix}\right]$.

1. *A matrix* $S \in \mathbb{R}^{2m\times 2m}$ *is called symplectic if* $SJS^T = J$.

2. *A matrix* $U \in \mathbb{R}^{2m\times 2m}$ *is called orthogonal symplectic if* $U^TU = I$ *and* $UJU^T = J$.

3. *A matrix* $A \in \mathbb{R}^{2m\times 2m}$ *is called Hamiltonian if* $JA = (JA)^T$.

The SVD-like decomposition (1.1) is closely related to the canonical forms of the real skew-symmetric matrix BJB^T and the real Hamiltonian matrix JB^TB. By (1.1) and the symplectic property of S, we have the *Schur-like form* for BJB^T,

$$BJB^T = Q \left[\begin{array}{cc|cc} 0 & 0 & \Sigma^2 & 0 \\ 0 & 0 & 0 & 0 \\ \hline -\Sigma^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] Q^T, \tag{1.2}$$

*Received by the editors June 22, 2002; accepted for publication (in revised form) by G. H. Golub May 30, 2004; published electronically May 6, 2005. This research was partially supported by NSF grant EPS-9874732, matching support from the state of Kansas, and the University of Kansas General Research Fund allocation 2301717.

http://www.siam.org/journals/simax/26-4/41052.html

†Department of Mathematics, University of Kansas, Lawrence, KS 66045 (xu@math.ukans.edu).

and the *structured canonical form* for JB^TB,

$$JB^TB = S \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & \Sigma^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -\Sigma^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] S^{-1} =: S\Gamma S^{-1}. \tag{1.3}$$

(Note that the condensed matrix Γ is still Hamiltonian.) In fact, let $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$. With appropriate permutations, (1.2) can be transformed to the real Schur form of BJB^T,

$$\mathrm{diag}\left(\left[\begin{array}{cc} 0 & \sigma_1^2 \\ -\sigma_1^2 & 0 \end{array}\right], \ldots, \left[\begin{array}{cc} 0 & \sigma_p^2 \\ -\sigma_p^2 & 0 \end{array}\right], \underbrace{0, \ldots, 0}_{n-2p} \right),$$

and (1.3) can be transformed to the real Jordan canonical form of JB^TB,

$$\mathrm{diag}\left(\left[\begin{array}{cc} 0 & \sigma_1^2 \\ -\sigma_1^2 & 0 \end{array}\right], \ldots, \left[\begin{array}{cc} 0 & \sigma_p^2 \\ -\sigma_p^2 & 0 \end{array}\right], \underbrace{\left[\begin{array}{cc} 0 & 0 \\ -1 & 0 \end{array}\right], \ldots \left[\begin{array}{cc} 0 & 0 \\ -1 & 0 \end{array}\right]}_{q}, \underbrace{0, \ldots, 0}_{2(m-p-q)} \right).$$

In this paper we will develop a numerical method to compute the SVD-like decomposition (1.1). Our main goal is to use it to compute the structured canonical form (1.3) of the Hamiltonian matrices JB^TB.

The eigenvalue problem of such Hamiltonian matrices has a variety of applications. One example is the linear Hamiltonian system [19]

$$\dot{x}(t) = JAx(t), \qquad x(0) = x_0,$$

where $A \in \mathbb{R}^{2m \times 2m}$ is real symmetric positive definite. The solution of such a Hamiltonian system satisfies

$$x^T(t)Ax(t) = x_0^T A x_0 \quad \forall t \geq 0. \tag{1.4}$$

This shows one fundamental principle of the Hamiltonian system, the conservation law. The solution $x(t)$ can be computed by using the structured canonical form of the Hamiltonian matrix JA. Since A is positive definite, one can compute the factorization $A = B^TB$, say, the Cholesky factorization. After having computed the SVD-like decomposition of B, one has

$$JA = S \left[\begin{array}{cc} 0 & \Sigma^2 \\ -\Sigma^2 & 0 \end{array}\right] S^{-1} =: S\Gamma S^{-1}.$$

(Note that Γ is slightly different from that in (1.3), because here A is nonsingular.) The solution can be computed by the following formula:

$$x(t) = Se^{\Gamma t}S^{-1}x_0.$$

It is easily verified that for any t, $e^{\Gamma t}$ is symplectic. If S is exactly symplectic, then

one can verify that

$$x^T(t)Ax(t) = x^T(t)J^{-1}(JA)x(t) = (Se^{\Gamma t}S^{-1}x_0)^T J^{-1}(S\Gamma S^{-1})Se^{\Gamma t}S^{-1}x_0 = x_0^T Ax_0.$$

Numerically, for the solution $x(t)$ to obey the conservation law (1.4), one needs to compute the eigenvalues of JA and the symplectic matrix S accurately.

Another example involves the gyroscopic system [8, 13, 17]

$$\ddot{q} + C\dot{q} + Gq = 0, \qquad q(0) = q_0, \quad \dot{q}(0) = q_1.$$

where $G \in \mathbb{R}^{m\times m}$ is symmetric and $C \in \mathbb{R}^{m\times m}$ is skew-symmetric. This system is related to the eigenvalue problem of the matrix

$$F = \begin{bmatrix} -C & -G \\ I & 0 \end{bmatrix} = \begin{bmatrix} -C & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & G \end{bmatrix}.$$

When G is positive semidefinite it has a full rank factorization $G = LL^T$. By using the equality

$$\begin{bmatrix} -C & -I \\ I & 0 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}C & I \\ I & 0 \end{bmatrix} J \begin{bmatrix} \frac{1}{2}C & I \\ I & 0 \end{bmatrix},$$

F is similar to the Hamiltonian matrix

$$J \begin{bmatrix} \frac{1}{2}C & I \\ I & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & LL^T \end{bmatrix} \begin{bmatrix} -\frac{1}{2}C & I \\ I & 0 \end{bmatrix} = J \begin{bmatrix} -\frac{1}{2}C & I \\ L^T & 0 \end{bmatrix}^T \begin{bmatrix} -\frac{1}{2}C & I \\ L^T & 0 \end{bmatrix}.$$

Then the eigenvalue problem of F can be solved by computing the SVD-like decomposition of $\left[\begin{smallmatrix} -\frac{1}{2}C & I \\ L^T & 0 \end{smallmatrix}\right]$.

The eigenvalues of JB^TB can be computed in many ways. For example, one can use the structure preserving method [2, 3]. Since the eigenvalues of JB^TB and BJB^T are the same, a more efficient and reliable way is to use the QR method or the Jacobi method (e.g., [15, 11]) to compute the eigenvalues of the skew-symmetric matrix BJB^T. A common problem of these methods is that they cannot compute the symplectic matrix S simultaneously. Another problem is that the methods work on the explicit matrix product JB^TB or BJB^T. The method that will be developed in this paper computes the SVD-like decomposition of B. So it computes both the eigenvalues of JB^TB and the matrix S simultaneously. Moreover, since it works only on the factor B, the eigenvalues of JB^TB can be computed more accurately. This trick is not new; see, e.g., [9, 14]. It has been also used to develop other singular value and eigenvalue methods [5, 12, 1, 10].

The basic idea of the method is introduced in section 2, and the reduction and iteration processes are described in section 3. In these two sections we focus on a matrix B with BJB^T nonsingular. A detail reduction process for a general matrix B is presented in section 4. The first order error bound for the computed eigenvalues is provided in section 5. Numerical examples are given in section 6. The conclusion is given in section 7.

In this paper $\|\cdot\|$ denotes the spectral norm.

2. The basic idea. We use the following procedure to compute an SVD-like decomposition. First compute a condensed form of B by using only orthogonal transformations. Then use the condensed form to construct the SVD-like decomposition. The method for computing the condensed form is actually the implicit version of the QR-like method for the real skew-symmetric matrix BJB^T. In order to describe the method in a simple way, in this and the next sections for a matrix B under consideration we assume that BJB^T is nonsingular. With this assumption B is necessarily of full row rank and has an even number of rows. A detailed process for a general matrix B will be presented in section 4.

For a nonsingular skew-symmetric matrix $K \in \mathbb{R}^{2p\times 2p}$ one can apply the QR-like algorithm to compute its Schur form. The algorithm consists of two steps. First apply a reduction procedure (see section 3) to K to obtain a bidiagonal-like form

$$Q_1^T K Q_1 = \begin{bmatrix} 0 & T \\ -T^T & 0 \end{bmatrix}, \tag{2.1}$$

where Q_1 is real orthogonal and $T \in \mathbb{R}^{p\times p}$ is upper bidiagonal. Then apply the QR-like SVD iteration to T to compute the SVD

$$T = Z_1 \Delta Z_2^T,$$

where Z_1, Z_2 are real orthogonal and Δ is positive diagonal. Let $Q = Q_1 \left[\begin{smallmatrix} Z_1 & 0 \\ 0 & Z_2 \end{smallmatrix}\right]$. Then we have the Schur-like form

$$Q^T K Q = \begin{bmatrix} 0 & \Delta \\ -\Delta & 0 \end{bmatrix}.$$

When $K = BJB^T$, we will develop an implicit version of the method by operating only on the factor B. Since $(Q^T BU)J(Q^T BU)^T = Q^T(BJB^T)Q$ for any orthogonal symplectic matrix U, we intend to determine an orthogonal matrix Q and an orthogonal symplectic matrix U such that $R = Q^T BU$ is block upper triangular and the product RJR^T has the Schur-like form. Similarly we need two steps to compute such a decomposition. We first determine an orthogonal matrix Q_1 and an orthogonal symplectic matrix U_1 such that

$$Q_1^T B U_1 = \begin{bmatrix} B_1 & B_2 \\ 0 & B_3 \end{bmatrix},$$

where $B_1, B_2, B_3 \in \mathbb{R}^{p\times m}$, and

$$Q_1^T BJB^T Q_1 = \begin{bmatrix} B_1B_2^T - B_2B_1^T & B_1B_3^T \\ -B_3B_1^T & 0 \end{bmatrix} = \begin{bmatrix} 0 & B_1B_3^T \\ -B_3B_1^T & 0 \end{bmatrix} \tag{2.2}$$

has the bidiagonal-like form (2.1). (This implies that $B_1B_2^T = B_2B_1^T$ and $B_1B_3^T$ is upper bidiagonal.) We then apply an implicit version of the QR-like SVD iteration to $B_1B_3^T$, to obtain

$$R_1 = Z_1^T B_1 W, \qquad R_3 = Z_2^T B_3 W, \tag{2.3}$$

where Z_1, Z_2, W are orthogonal and $R_1R_3^T = \Delta$ is positive diagonal. Let $Q = Q_1 \left[\begin{smallmatrix} Z_1 & 0 \\ 0 & Z_2 \end{smallmatrix}\right]$ and $U = U_1 \left[\begin{smallmatrix} W & 0 \\ 0 & W \end{smallmatrix}\right]$ (which is orthogonal symplectic). Then

$$R = Q^T BU = \begin{bmatrix} R_1 & R_2 \\ 0 & R_3 \end{bmatrix}, \qquad R_2 = Z_1^T B_2 W.$$

By (2.2) and (2.3), we have $Q^T(BJB^T)Q = RJR^T = \left[\begin{smallmatrix} 0 & \Delta \\ -\Delta & 0 \end{smallmatrix}\right]$.

The most condensed form that we can compute for B is

$$R = Q^T BU = \left[\begin{array}{cc|cc} R_{11} & R_{12} & R_{13} & R_{14} \\ \hline 0 & 0 & R_{23} & 0 \end{array}\right] =: \begin{bmatrix} R_1 & R_2 \\ 0 & R_3 \end{bmatrix}, \tag{2.4}$$

where $R_{11}, R_{23} \in \mathbb{R}^{p\times p}$, R_{11} is upper triangular, R_{23} is lower triangular, and $R_{11}R_{23}^T =: \Delta$ is positive diagonal. The detailed procedure will be presented in the next section. Let $\Delta = \operatorname{diag}(\delta_1, \ldots, \delta_p)$. After having obtained such a decomposition the eigenvalues of BJB^T and JB^TB are simply $\pm i\delta_1, \ldots, \pm i\delta_p$. Define $\Sigma = \sqrt{\Delta}$. The symplectic matrix S in the SVD-like decomposition can be computed by the formula

$$U \left[\begin{array}{cc|cc} R_{23}^T\Sigma^{-1} & -(R_{23}^T\Sigma^{-1})(R_{12}^T\Sigma^{-1})^T & -R_{13}^T\Sigma^{-1} & -(R_{23}^T\Sigma^{-1})(R_{14}^T\Sigma^{-1})^T \\ 0 & I & -R_{14}^T\Sigma^{-1} & 0 \\ \hline 0 & 0 & R_{11}^T\Sigma^{-1} & 0 \\ 0 & 0 & R_{12}^T\Sigma^{-1} & I \end{array}\right], \tag{2.5}$$

and the SVD-like decomposition of B is

$$Q^T BS = \begin{array}{c} \\ p \\ p \end{array} \begin{array}{c} \begin{array}{cccc} p & m-p & p & m-p \end{array} \\ \begin{pmatrix} \Sigma & 0 & 0 & 0 \\ 0 & 0 & \Sigma & 0 \end{pmatrix} \end{array}. \tag{2.6}$$

Note this is the decomposition only in the case that BJB^T is nonsingular.

The method is summarized by the following algorithm.

ALGORITHM. Given a real matrix $B \in \mathbb{R}^{2p\times 2m}$ with BJB^T nonsingular, the algorithm computes the eigenvalues of JB^TB and BJB^T or the SVD-like decomposition (2.6).

Step 1. Determine the orthogonal matrix Q_1 and the orthogonal symplectic matrix U_1 such that

$$Q_1^T BU_1 = \left[\begin{array}{cc|cc} B_{11} & B_{12} & B_{13} & B_{14} \\ \hline 0 & 0 & B_{23} & 0 \end{array}\right] =: \begin{bmatrix} B_1 & B_2 \\ 0 & B_3 \end{bmatrix}, \tag{2.7}$$

where $B_{11}, B_{23} \in \mathbb{R}^{p\times p}$, B_{11} is upper triangular, B_{23} is lower triangular, $B_{11}B_{23}^T$ is upper bidiagonal, and $B_1B_2^T = B_2B_1^T$.

Step 2. Determine the orthogonal matrices Z_1, Z_2, W such that

$$R_{11} = Z_1^T B_{11} W, \qquad R_{23} = Z_2^T B_{23} W, \tag{2.8}$$

where R_{11} is upper triangular, R_{23} is lower triangular, and

$$R_{11}R_{23}^T = \operatorname{diag}(\delta_1, \ldots, \delta_p) =: \Delta$$

is positive diagonal.

Step 3. If only the eigenvalues of JB^TB or BJB^T are required, compute the nonzero eigenvalues $\pm i\delta_1, \ldots, \pm i\delta_p$ and stop. If the decomposition (2.6) is required, go to Step 4.

Step 4.

(a) Update $Q = Q_1\left[\begin{smallmatrix} Z_1 & 0 \\ 0 & Z_2 \end{smallmatrix}\right]$, $U = U_1 \operatorname{diag}(W, I, W, I)$, and

$$R = \left[\begin{array}{cc|cc} R_{11} & R_{12} & R_{13} & R_{14} \\ \hline 0 & 0 & R_{23} & 0 \end{array}\right],$$

where $R_{12} = Z_1^T B_{12}$, $R_{13} = Z_1^T B_{13} W$, and $R_{14} = Z_1^T B_{14}$.

(b) Compute $\Sigma = \sqrt{\Delta}$.

(c) Use the formula (2.5) to compute S.

3. Reduction and iteration. We need the following elementary matrices in our algorithm.

1. *Set of Householder matrices:*

$$\mathcal{H}(\mathcal{I}) = \{H = I_n - 2uu^T/u^Tu \mid u \in \mathbb{R}^n,\ u_j = 0,\ \forall j \notin \mathcal{I}\},$$

where $\mathcal{I}$ is a subset of $\{1, \ldots, n\}$ giving the range of the columns and rows that H operates on.

2. *Set of Givens matrices:*

$$\mathcal{G}(i,j) = \{G \mid G = I_n - (1-\alpha)(e_ie_i^T + e_je_j^T) + \beta(e_ie_j^T - e_je_i^T),\ \alpha^2 + \beta^2 = 1\}.$$

3. *Set of symplectic Householder matrices:*

$$\mathcal{H}^s(\mathcal{I}) = \left\{ H_s \mid H_s := \begin{bmatrix} H & 0 \\ 0 & H \end{bmatrix},\ H \in \mathcal{H}(\mathcal{I}) \right\}.$$

4. *Sets of symplectic Givens matrices:*
   (a) $\mathcal{G}_1^s(k) = \{G_s \mid G_s \in \mathcal{G}(k, n+k) \subset \mathbb{R}^{2n\times 2n}\}$.
   (b) $\mathcal{G}_2^s(i,j) = \{G_s = \left[\begin{smallmatrix} G & 0 \\ 0 & G \end{smallmatrix}\right] \mid G \in \mathcal{G}(i,j)\}$.
   (c) $\mathcal{G}_3^s(i,j) = \left\{ G_s \left| \begin{array}{l} G_s = I_{2n} - (1-\alpha)(e_ie_i^T + e_je_j^T + e_{n+i}e_{n+i}^T + e_{n+j}e_{n+j}^T) \\ \quad +\beta(e_ie_{n+j}^T + e_je_{n+i}^T - e_{n+j}e_i^T - e_{n+i}e_j^T),\ \alpha^2+\beta^2 = 1 \end{array} \right. \right\}$,

where $1 \le i < j \le n$.

5. *Sets of symplectic permutations:*
   (a) $\mathcal{P}_1^s = \{\left[\begin{smallmatrix} P & 0 \\ 0 & P \end{smallmatrix}\right] \mid P$ is a permutation$\}$.
   (b) $\mathcal{P}_2^s(k) = \{P_s \mid P_s = I_{2n} - (e_ke_k^T + e_{n+k}e_{n+k}^T) + (e_ke_{n+k}^T - e_{n+k}e_k^T)\}$.

In the algorithm Steps 3 and 4 are simple. So we consider only the implementations for Step 1 and 2.

3.1. Implicit bidiagonal-like reduction. We use the following displays with a 6×8 matrix B to illustrate the reduction process. In the displays, "0" and "x" denote a zero and an arbitrary element, respectively. Note that our goal is to reduce B to a condensed form (2.7) such that the explicit product BJB^T has a bidiagonal-like form (2.1).

At the first stage we reduce the columns and rows 1 and 4 of BJB^T implicitly. For this we first perform three orthogonal symplectic transformations $U_{1,1}, V_1, U_{1,2}$ successively, where $U_{1,1}, U_{1,2} \in \mathcal{H}^s(1:4)$ and $V_1 \in \mathcal{G}_1^s(1)$, on the columns of B to annihilate $B(4, 2:4)$, $B(4,1)$, and $B(4, 6:8)$:[1]

$$\left[\begin{array}{cccc|cccc} x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ \hline 0 & 0 & 0 & 0 & x & 0 & 0 & 0 \\ x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \end{array}\right].$$

We then perform a Householder transformation $H_{1,1} \in \mathcal{H}(1:3, 5:6)$ on the rows of

[1] Here we use the MATLAB forms to denote the entries, rows, and columns of a matrix.

B to annihilate $B(2:3,1)$ and $B(5:6,1)$:

$$\left[\begin{array}{cccc|cccc} x & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x \\ \hline 0 & 0 & 0 & 0 & x & 0 & 0 & 0 \\ 0 & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x \end{array}\right].$$

Now the product $B(JB^T)$ has the form

$$\left[\begin{array}{cccc|cccc} x & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x \\ \hline 0 & 0 & 0 & 0 & x & 0 & 0 & 0 \\ 0 & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x \end{array}\right] \left[\begin{array}{ccc|ccc} x & x & x & x & x & x \\ x & x & x & 0 & x & x \\ x & x & x & 0 & x & x \\ \hline x & 0 & 0 & 0 & 0 & 0 \\ x & x & x & 0 & x & x \\ x & x & x & 0 & x & x \\ x & x & x & 0 & x & x \end{array}\right] = \left[\begin{array}{ccc|ccc} 0 & x & x & x & x & x \\ x & 0 & x & 0 & x & x \\ x & x & 0 & 0 & x & x \\ \hline x & 0 & 0 & 0 & 0 & 0 \\ x & x & x & 0 & 0 & x \\ x & x & x & 0 & x & 0 \end{array}\right].$$

(Since BJB^T is skew-symmetric, its diagonal elements are zero.) We still need to reduce the first column and row of BJB^T. For this we have to form the first column (but not the whole product) of BJB^T explicitly, which has the pattern

$$y_1 = \left[\begin{array}{cccccc} 0 & x & x & x & x & x \end{array}\right]^T.$$

Determine a Householder matrix $H_{1,2} \in \mathcal{H}(2:3,5:6)$ such that

$$H_{1,2}y_1 = \left[\begin{array}{cccccc} 0 & 0 & 0 & x & x & 0 \end{array}\right]^T.$$

Premultiply B by $H_{1,2}$. Since $H_{1,2}$ does not work on rows 1 and 4, it does not change the pattern of B. After this transformation

$$B = \left[\begin{array}{cccc|cccc} x & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x \\ \hline 0 & 0 & 0 & 0 & x & 0 & 0 & 0 \\ 0 & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x \end{array}\right], \qquad BJB^T = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & x & x & 0 \\ 0 & 0 & x & 0 & x & x \\ 0 & x & 0 & 0 & x & x \\ \hline x & 0 & 0 & 0 & 0 & 0 \\ x & x & x & 0 & 0 & x \\ 0 & x & x & 0 & x & 0 \end{array}\right].$$

The second stage is similar. We reduce the columns and rows 2 and 5 of BJB^T. We first perform transformations $U_{2,1}, V_2, U_{2,2}$, where $U_{2,1}, U_{2,2} \in \mathcal{H}^s(2:4)$ and $V_2 \in \mathcal{G}_1^s(2)$, on the columns of B to annihilate $B(5,3:4)$, $B(5,2)$, and $B(5,7:8)$. Then perform a Householder transformation $H_{2,1} \in \mathcal{H}(2:3,6)$ on the rows of B to annihilate $B(3,2)$ and $B(6,2)$. Next we determine a Householder transformation $H_{2,2} \in \mathcal{H}(3,6)$ from the vector

$$y_2 = (BJB^T)(:,2) = \left[\begin{array}{cccccc} 0 & 0 & x & 0 & x & x \end{array}\right]^T,$$

such that

$$H_{2,2}y_2 = \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & x & x \end{array}\right]^T.$$

Premultiplying B by $H_{2,2}$,

$$B = \left[\begin{array}{cccc|cccc} x & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x \\ 0 & 0 & x & x & x & x & x & x \\ \hline 0 & 0 & 0 & 0 & x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x & x & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x \end{array}\right],$$

and

$$BJB^T = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & x & x & 0 \\ 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & x \\ \hline x & 0 & 0 & 0 & 0 & 0 \\ x & x & 0 & 0 & 0 & 0 \\ 0 & x & x & 0 & 0 & 0 \end{array}\right].$$

Now the product BJB^T is in the bidiagonal-like form.

At the third stage we perform transformations $U_{3,1}, V_3, U_{3,2}$, where $U_{3,1}, U_{3,2} \in \mathcal{H}^s(3:4)$ and $V_3 \in \mathcal{G}_1^s(3)$, on the columns of B to annihilate $B(6,4)$, $B(6,3)$, and $B(6,8)$:

$$B = \left[\begin{array}{ccc|c|ccc|c} x & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x \\ 0 & 0 & x & x & x & x & x & x \\ \hline 0 & 0 & 0 & 0 & x & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & x & x & 0 & 0 \\ 0 & 0 & 0 & 0 & x & x & x & 0 \end{array}\right].$$

We have got the form (2.7). Note that the symplectic transformations performed at the last stage do not change the bidiagonal-like form of BJB^T.

3.2. Implicit QR-like SVD iteration. We will give the implicit version of the implicit shift QR-like SVD iteration [11, sect. 8.6] on $B_{11}B_{23}^T$. For a technical reason related to decoupling and deflation, before iteration we transform B_{23} to a lower Hessenberg form such that $B_{11}B_{23}^T$ is lower bidiagonal. The transformations can be performed as follows. For $j = 1, \ldots, p-1$, we construct a sequence of Givens matrices $G_j \in \mathcal{G}(j, j+1)$ such that $(B_{11}B_{23}^T)G_1 \cdots G_{p-1}$ becomes lower bidiagonal. (To construct G_j we need to compute $(B_{11}B_{23}^T)(j, j:j+1)$.) Update $B_{23} := G_{p-1}^T \cdots G_1^T B_{23}$. Then B_{23} becomes lower Hessenberg. B_{11} is still upper triangular, but now $B_{11}B_{23}^T$ is lower bidiagonal.

When some diagonal or subdiagonal elements of $B_{11}B_{23}^T$ are zero we can decouple it into smaller unreduced lower bidiagonal blocks. With the assumption that BJB^T is nonsingular all diagonal elements of $B_{11}B_{23}^T$ are nonzero. This is obvious from the factorization $BJB^T = Q_1 \left[\begin{smallmatrix} 0 & B_{11}B_{23}^T \\ -B_{23}^T B_{11} & 0 \end{smallmatrix}\right] Q_1^T$. Moreover, B_{11} is nonsingular. Hence its diagonal elements are nonzero. The jth subdiagonal element of $B_{11}B_{23}^T$ has the form[2] $B_{11}(j+1, j+1)B_{23}(j, j+1)$. Because $B_{11}(j+1, j+1) \neq 0$, the jth subdiagonal

[2]This is why we transform B_{23} to a lower Hessenberg form. In the upper bidiagonal case the jth superdiagonal element of $B_{11}B_{23}^T$ is in a dot product form $B_{11}(j,j)B_{23}(j+1,j) + B_{11}(j,j+1)B_{23}(j+1,j+1)$. It may happen that this dot product is small but all four elements are not small. When this happens, we have the difficulty of doing the decoupling or deflation.

element of $B_{11}B_{23}^T$ is zero if and only if the jth superdiagonal element of B_{23} is zero. With this observation, in practice when some superdiagonal elements of B_{23} are zero or suitably small we set them to be zero and decouple $B_{11}B_{23}^T$ into smaller unreduced lower bidiagonal blocks. We then perform the following implicit version of the QR-like SVD iterations to each pair of small diagonal blocks from B_{11} and B_{23} corresponding to each unreduced block in $B_{11}B_{23}^T$ to compute (2.8). The criterion for decoupling or deflation that we use is

$$|B_{23}(j,j+1)| \le \varepsilon(|B_{23}(j,j)| + |B_{23}(j+1,j)| + |B_{23}(j+1,j+1)|), \tag{3.1}$$

where ε is the machine precision. With this criterion decoupling or deflation will cause an error in B of order $\varepsilon\|B\|$.

We use the matrices B_{11}, B_{23} with size 4×4 to illustrate one step of iteration. Initially B_{11} is upper triangular, B_{23} is lower Hessenberg, and $B_{11}B_{23}^T$ is lower bidiagonal. Without loss of generality we assume that $B_{11}B_{23}^T$ is unreduced. Let $\delta > 0$ be a shift.[3] Let A be the leading 2×2 principal submatrix of $B_{11}B_{23}^T B_{23}B_{11}^T$. We first determine a Givens matrix $G_1 \in \mathcal{G}(1,2)$, in which the leading 2×2 principal submatrix is a Givens rotation that transforms $A - \delta I$ to an upper triangular form. Perform G_1 on the rows of B_{11}:

$$B_{11} = \begin{bmatrix} x & x & x & x \\ \otimes & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix},$$

where "$\otimes$" denotes an unwanted nonzero element. Now the product becomes

$$B_{11}B_{23}^T = \begin{bmatrix} x & \otimes & 0 & 0 \\ x & x & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & x \end{bmatrix}.$$

Perform a Givens transformation $W_1 \in \mathcal{G}(1,2)$ on the columns of B_{11} to annihilate $B_{11}(2,1)$ and perform it also on the columns of B_{23}:

$$B_{11} = \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix}, \qquad B_{23} = \begin{bmatrix} x & x & 0 & 0 \\ x & x & x & 0 \\ x & x & x & x \\ x & x & x & x \end{bmatrix}.$$

This transformation does not change the pattern of $B_{11}B_{23}^T$. Next we determine a Givens matrix $S_1 \in \mathcal{G}(1,2)$ to annihilate $(B_{11}B_{23}^T)(1,2)$. (Again, in order to construct S_1 we need to compute $(B_{11}B_{23}^T)(1,1:2)$.) Perform S_1 on the rows of B_{23}:

$$B_{23} = \begin{bmatrix} x & x & \otimes & 0 \\ x & x & x & 0 \\ x & x & x & x \\ x & x & x & x \end{bmatrix},$$

[3] We actually use the Wilkinson shift, one of the eigenvalues of the tailing 2×2 principal submatrix of $B_{11}B_{23}^T B_{23}B_{11}^T$.

and now

$$B_{11}B_{23}^T = \begin{bmatrix} x & 0 & 0 & 0 \\ x & x & 0 & 0 \\ \otimes & x & x & 0 \\ 0 & 0 & x & x \end{bmatrix}.$$

To annihilate $(B_{11}B_{23}^T)(3,1)$ we first perform a Givens transformation $W_2 \in \mathcal{G}(2,3)$ on the columns of B_{23} to annihilate $B_{23}(1,3)$. Perform W_2 also on the columns of B_{11}:

$$B_{11} = \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & \otimes & x & x \\ 0 & 0 & 0 & x \end{bmatrix}, \qquad B_{23} = \begin{bmatrix} x & x & 0 & 0 \\ x & x & x & 0 \\ x & x & x & x \\ x & x & x & x \end{bmatrix}.$$

Then we perform a Givens transformation $G_2 \in \mathcal{G}(2,3)$ on the rows of B_{11} to annihilate $B_{11}(3,2)$:

$$B_{11} = \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix}.$$

At this stage

$$B_{11}B_{23}^T = \begin{bmatrix} x & 0 & 0 & 0 \\ x & x & \otimes & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & x \end{bmatrix}.$$

So $(B_{11}B_{23}^T)(3,1)$ has been annihilated and the bulge has been chased to the $(2,3)$ place. In a similar way we can chase the bulge down-rightwards until it disappears. The rest of the reductions are illustrated by the following displays, where B_{11} and B_{23} are displayed simultaneously, the Givens transformation $G_j \in \mathcal{G}(j, j+1)$ operates only on the rows of B_{11}, $S_j \in \mathcal{G}(j, j+1)$ operates only on the rows of B_{23}, and $W_j \in \mathcal{G}(j, j+1)$ operates on the columns of both B_{11} and B_{23}.

$$\begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix}, \quad \begin{bmatrix} x & x & 0 & 0 \\ x & x & x & 0 \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \xrightarrow{S_2} \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix}, \quad \begin{bmatrix} x & x & 0 & 0 \\ x & x & x & \otimes \\ x & x & x & x \\ x & x & x & x \end{bmatrix}$$

$$\xrightarrow{W_3} \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & \otimes & x \end{bmatrix}, \quad \begin{bmatrix} x & x & 0 & 0 \\ x & x & x & 0 \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \xrightarrow{G_3} \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix}, \quad \begin{bmatrix} x & x & 0 & 0 \\ x & x & x & 0 \\ x & x & x & x \\ x & x & x & x \end{bmatrix}$$

$$\xrightarrow{S_3} \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix}, \quad \begin{bmatrix} x & x & 0 & 0 \\ x & x & x & 0 \\ x & x & x & x \\ x & x & x & x \end{bmatrix}.$$

We have finished one step of iteration.

We now check the superdiagonal elements of B_{23}. If some of them satisfy (3.1), we replace them by zero and decouple or deflate $B_{11}B_{23}^T$. We then run another step of iteration on B_{11} and B_{23} or a pair of diagonal blocks from them. Repeat the iterations and finally B_{23} becomes lower triangular and we have (2.8).

The algorithm costs about two to three times as much as the QR-like algorithm applied to the explicit product BJB^T.

4. General case. For a general matrix $B \in \mathbb{R}^{n\times 2m}$ additional work needs to be done. If $\operatorname{rank} B < n$, initially we need to compute a factorization

$$B = Q_0 \begin{bmatrix} B_0 \\ 0 \end{bmatrix}, \tag{4.1}$$

where Q_0 is orthogonal and B_0 is of full row rank. This can be done by the QR factorization with the column pivoting method (see [6]), the rank-revealing QR (see [7]), or the SVD algorithm (see [11, sect. 8.6]).

Next we apply the reduction process to B_0. But now we have to modify the above reduction process slightly. The reason is that even if B_0 is of full row rank, the product $B_0JB_0^T$ may be singular. In this case at certain stages of reductions some diagonal elements of block B_{11} or B_{23} will be zero and we need to deflate the zero eigenvalues of $B_0JB_0^T$. Because of this, we have to reduce matrix B_0 to a more generalized condensed form

$$Q_2^T B_0 U_2 = \begin{array}{c} \\ p \\ q \\ p \end{array} \begin{array}{c} \begin{array}{cccccc} p & q & m-p-q & p & q & m-p-q \end{array} \\ \begin{pmatrix} B_{11} & B_{12} & B_{13} & B_{14} & B_{15} & B_{16} \\ 0 & B_{22} & 0 & B_{24} & 0 & 0 \\ 0 & 0 & 0 & B_{34} & 0 & 0 \end{pmatrix} \end{array}, \tag{4.2}$$

where Q_2 is orthogonal, U_2 is orthogonal symplectic, B_{11}, B_{22} are nonsingular and upper triangular, B_{34} is nonsingular and lower triangular, $B_{11}B_{34}^T$ is upper bidiagonal, and

$$Q_2^T B_0 J B_0 Q_2 = \begin{array}{c} \\ p \\ q \\ p \end{array} \begin{array}{c} \begin{array}{ccc} p & q & p \end{array} \\ \begin{pmatrix} 0 & 0 & B_{11}B_{34}^T \\ 0 & 0 & 0 \\ -B_{34}B_{11}^T & 0 & 0 \end{pmatrix} \end{array}. \tag{4.3}$$

The reduction procedure will be illustrated below. We then apply the same iteration procedure described in subsection 3.2 to B_{11}, B_{34} to compute

$$R_{11} = Z_1^T B_{11} W, \qquad R_{34} = Z_2^T B_{34} W,$$

where Z_1, Z_2, W are orthogonal, R_{11} is upper triangular, R_{34} is lower triangular, and $\Delta := R_{11}R_{34}^T$ is positive diagonal. Similarly, combining them with (4.2) and (4.1) we can determine the orthogonal matrix Q and the orthogonal symplectic matrix U to obtain the generalized version of (2.4),

$$Q^T B U = \begin{array}{c} \\ p \\ q \\ p \\ n-2p-q \end{array} \begin{array}{c} \begin{array}{cccccc} p & q & m-p-q & p & q & m-p-q \end{array} \\ \begin{pmatrix} R_{11} & R_{12} & R_{13} & R_{14} & R_{15} & R_{16} \\ 0 & R_{22} & 0 & R_{24} & 0 & 0 \\ 0 & 0 & 0 & R_{34} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

Let $\Sigma = \sqrt{\Delta}$. The symplectic matrix S can be computed by the formula

$$U \left[\begin{array}{ccc|ccc} X & -X(R_{12}^T\Sigma^{-1})^T & -X(R_{13}^T\Sigma^{-1})^T & -R_{14}^T\Sigma^{-1} & -X(R_{15}^T\Sigma^{-1})^T & -X(R_{16}^T\Sigma^{-1})^T \\ 0 & I & 0 & -R_{15}^T\Sigma^{-1} & 0 & 0 \\ 0 & 0 & I & -R_{16}^T\Sigma^{-1} & 0 & 0 \\ \hline 0 & 0 & 0 & R_{11}^T\Sigma^{-1} & 0 & 0 \\ 0 & 0 & 0 & R_{12}^T\Sigma^{-1} & I & 0 \\ 0 & 0 & 0 & R_{13}^T\Sigma^{-1} & 0 & I \end{array}\right],$$

where $X = R_{34}^T\Sigma^{-1}$. Finally we have the SVD-like decomposition

$$Q^T BS = \begin{array}{c} \\ p \\ q \\ p \\ n-2p-q \end{array} \begin{array}{c} \begin{array}{cccccc} p & q & m-p-q & p & q & m-p-q \end{array} \\ \begin{pmatrix} \Sigma & 0 & 0 & 0 & 0 & 0 \\ 0 & R_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Sigma & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

(If necessary one can multiply the symplectic matrix $\mathrm{diag}(I, R_{22}^{-1}, I; I, R_{22}^T, I)$ from the right to replace R_{22} by I.)

In the following we will show the reduction procedure for computing the condensed form (4.2). The procedure consists of two steps. In step 1 we will reduce B_0 to

$$Q_2^T B_0 \tilde{U}_2 = \begin{array}{c} \\ p \\ q \\ p \end{array} \begin{array}{c} \begin{array}{cccccc} p & m-p-q & q & p & m-p-q & q \end{array} \\ \begin{pmatrix} B_{11} & B_{12} & B_{13} & B_{14} & B_{15} & 0 \\ 0 & 0 & B_{23} & 0 & 0 & 0 \\ 0 & 0 & B_{33} & B_{34} & 0 & 0 \end{pmatrix} \end{array}, \tag{4.4}$$

where Q_2 is orthogonal, $\tilde{U}_2$ is orthogonal symplectic, B_{11}, B_{23} are nonsingular and upper triangular, and B_{34} is nonsingular and lower triangular, such that $Q_2^T(B_0 J B_0^T)Q_2$ has the bidiagonal-like form (4.3). In step 2 we will perform only orthogonal symplectic transformations on the columns to transform (4.4) to (4.2). Note that step 2 does not change the bidiagonal-like form of $Q_2^T(B_0 J B_0^T)Q_2$.

Let us describe step 1 in an inductive way. Suppose that at a certain stage we have reduced B_0 to

$$B_0 = \begin{array}{c} \\ p \\ q \\ r \end{array} \begin{array}{c} \begin{array}{cccccc} j & m-j-q & q & j & m-j-q & q \end{array} \\ \begin{pmatrix} B_{11} & B_{12} & B_{13} & B_{14} & B_{15} & 0 \\ 0 & 0 & B_{23} & 0 & 0 & 0 \\ 0 & B_{32} & B_{33} & B_{34} & B_{35} & 0 \end{pmatrix} \end{array}$$

$$= \begin{array}{c} \\ j \\ p-j \\ q \\ j \\ r-j \end{array} \begin{array}{c} \begin{array}{cccccc} j & m-j-q & q & j & m-j-q & q \end{array} \\ \begin{pmatrix} \blacksquare & \square & \square & \square & \square & 0 \\ 0 & \square & \square & \square & \square & 0 \\ 0 & 0 & \blacksquare & 0 & 0 & 0 \\ 0 & 0 & \square & \blacksquare & 0 & 0 \\ 0 & \square & \square & \square & \square & 0 \end{pmatrix} \end{array}, \tag{4.5}$$

where $\blacksquare$ denotes a square upper triangular block, $\blacksquare$ a square lower triangular block, and $\square$ a square block.

and

$$
B_0JB_0 = \begin{array}{c} \\ j \\ p-j \\ q \\ j \\ r-j \end{array}
\begin{array}{c}
\begin{array}{ccccc} j & p-j & q & j & r-j \end{array} \\
\begin{pmatrix}
0 & 0 & 0 & \diagdown & * \\
0 & \square & 0 & 0 & \square \\
0 & 0 & 0 & 0 & 0 \\
\diagdown & 0 & 0 & 0 & 0 \\
* & \square & 0 & 0 & \square
\end{pmatrix}
\end{array}, \tag{4.6}
$$

where $r = p$ or $p+1$, $j \le p$. (Initially we partition B_0 to the block form with $j = q = 0$, and $r = p$ if B_0 has $2p$ (even) rows or $r = p+1$ if B_0 has $2p+1$ (odd) rows.) Note that in (4.5) when $j = p = r$ it is just (4.4) and we are done.

If $j \le p$, we continue the reduction process. We first perform orthogonal symplectic transformations on the columns of B_0 to annihilate $B_0(p+q+j+1, j+1 : m-q)$ and $B_0(p+q+j+1, m+j+2 : 2m-q)$, and then perform a Householder transformation on the rows of B_0 to annihilate $B_0(j+2 : p, j+1)$ and $B_0(p+q+j+2 : p+q+r, j+1)$. After this step we have two cases.

(a) $B_0(j+1, j+1) \neq 0$. We determine another Householder matrix to annihilate the elements from $j+2$ to p and from $p+q+j+3$ to $p+q+r$ on the $(j+1)$th column/row of $B_0JB_0^T$. Premultiply B_0 by this Householder matrix. Then B_0 and $B_0JB_0^T$ again have the block forms (4.5) and (4.6), respectively, but $j := j+1$. We have done one step of regular reduction as in subsection 3.1.

(b) $B_0(j+1, j+1) = 0$. We need to deflate the zero eigenvalue of $B_0JB_0^T$. We have two subcases:

(b1) $r = p+1$ and

(b2) $r = p$.

For the first subcase the deflation is illustrated by a matrix with $j = 2$, $p = 4$, $r = 5$, $q = 2$, and $m = 8$:

$$
B_0 = \left[\begin{array}{cc|cccc|cc|cc|cccc|cc}
x & x & x & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \\
0 & x & x & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \\
0 & 0 & \mathbf{0} & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \\
0 & 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & x & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & x & x & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & x & x & x & x & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & x & x & x & x & x & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \\
0 & 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & 0 & 0
\end{array}\right]
$$

and

$$B_0JB_0^T = \left[\begin{array}{cccc|cc|cccccc} 0&0&0&0&0&0&x&x&0&0&0\\ 0&0&0&0&0&0&0&x&x&0&0\\ 0&0&0&x&0&0&0&0&\mathbf{0}&x&x\\ 0&0&x&0&0&0&0&0&0&x&x\\ \hline 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0\\ \hline x&0&0&0&0&0&0&0&0&0&0\\ x&x&0&0&0&0&0&0&0&0&0\\ 0&x&\mathbf{0}&0&0&0&0&0&0&0&0\\ 0&0&x&x&0&0&0&0&0&0&x\\ 0&0&x&x&0&0&0&0&0&x&0 \end{array}\right].$$

For the explicit product we can perform a sequence of Givens transformations $G_1 \in \mathcal{G}(8,9)$ and $G_2 \in \mathcal{G}(7,8)$ on both the columns and rows to annihilate $(B_0JB_0^T)(2,8)$, $(B_0JB_0^T)(1,7)$ and $(B_0JB_0^T)(8,2)$, $(B_0JB_0^T)(7,1)$. With repartitioning we again have the form (4.6) but with $q = 3$:

$$B_0JB_0^T = \left[\begin{array}{ccccc|cc|cccc} 0&0&0&0&0&0&0&x&x&0&0\\ 0&0&0&0&0&0&0&0&x&0&0\\ 0&0&0&x&0&0&0&0&0&x&x\\ 0&0&x&0&0&0&0&0&0&x&x\\ \hline 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0\\ \hline x&0&0&0&0&0&0&0&0&0&0\\ x&x&0&0&0&0&0&0&0&0&0\\ 0&0&x&x&0&0&0&0&0&0&x\\ 0&0&x&x&0&0&0&0&0&x&0 \end{array}\right]. \tag{4.7}$$

The corresponding implicit version is as follows. We first perform a sequence of the symplectic Givens transformations $U_1 \in \mathcal{G}_2^s(2,3)$, $U_2 \in \mathcal{G}_2^s(1,2)$ on the columns of B_0 to annihilate $B_0(2,2)$ and $B_0(1,1)$:

$$B_0 = \left[\begin{array}{cc|cccc|cc|cc|cccc|cc} 0&x&x&x&x&x&x&x&x&x&x&x&x&x&0&0\\ 0&0&x&x&x&x&x&x&x&x&x&x&x&x&0&0\\ 0&0&0&x&x&x&x&x&x&x&x&x&x&x&0&0\\ 0&0&0&x&x&x&x&x&x&x&x&x&x&x&0&0\\ \hline 0&0&0&0&0&0&x&x&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&0\\ \hline 0&0&0&0&0&0&x&x&x&\otimes&0&0&0&0&0&0\\ 0&0&0&0&0&0&x&x&x&x&\otimes&0&0&0&0&0\\ 0&0&0&0&0&0&x&x&x&x&x&0&0&0&0&0\\ 0&0&0&x&x&x&x&x&x&x&x&x&x&x&0&0\\ 0&0&0&x&x&x&x&x&x&x&x&x&x&x&0&0 \end{array}\right].$$

Then perform Givens transformations $G_1 \in \mathcal{G}(8,9)$ and $G_2 \in \mathcal{G}(7,8)$ on the rows of

B_0 to annihilate the unwanted elements $B_0(8,11)$ and $B_0(7,10)$:

$$B_0 = \left[\begin{array}{cc|cccc|cc|cc|cccc|cc} 0 & x & x & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \\ 0 & 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \\ 0 & 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & x & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & x & x & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x & x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x & x & x & x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \\ 0 & 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \end{array}\right].$$

Now by using the pattern of B_0 one can see that $B_0 J B_0^T$ has the form (4.7). To transform B_0 back to the block form (4.5) next we perform a symplectic permutation $P_1 \in \mathcal{P}_1^s$ to move the columns 1 and 9 of B_0 to columns 6 and 14, respectively. Then we perform a symplectic permutation $P_2 \in \mathcal{P}_2^s(6)$ to interchange the columns 6 and 14. With repartitioning,

$$B_0 = \left[\begin{array}{ccc|ccc|ccc|ccc|ccc} x & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 & 0 \\ 0 & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & x & x & x & x & 0 & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & x & x & x & x & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & x & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x & x & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & x & x & x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x & x & x & x & x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & x & x & x & x & 0 & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & x & x & x & x & 0 & 0 & 0 \end{array}\right].$$

Note that these permutations do not change the form of $B_0 J B_0^T$. To maintain the block B_{23} in upper triangular form we perform a row permutation to move row 7 to row 5:

$$B_0 = \left[\begin{array}{ccc|ccc|ccc|ccc|ccc} x & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 & 0 \\ 0 & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & x & x & x & x & 0 & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & x & x & x & x & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & x & x & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & x & x & x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x & x & x & x & x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & x & x & x & x & 0 & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & x & x & x & x & 0 & 0 & 0 \end{array}\right].$$

Then B_0 and $B_0 J B_0^T$ again have the forms (4.5) and (4.6), respectively, but now $r := r - 1$ and $q := q + 1$.

For the second subcase the reduction procedure is illustrated by a matrix with $j = 2$, $p = 5$, $r = 5$, $q = 1$, and $m = 8$:

$$B_0 = \left[\begin{array}{cc|ccccc|ccc|ccccc|c} x & x & x & x & x & x & x & x & x & x & x & x & x & x & 0 \\ 0 & x & x & x & x & x & x & x & x & x & x & x & x & x & 0 \\ 0 & 0 & \mathbf{0} & x & x & x & x & x & x & x & x & x & x & x & 0 \\ 0 & 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & 0 \\ 0 & 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & x & x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x & x & x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x & x & x & x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & 0 \\ 0 & 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & 0 \end{array}\right],$$

and

$$B_0 J B_0^T = \left[\begin{array}{ccccc|c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 & x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & x & 0 & 0 \\ 0 & 0 & 0 & x & x & 0 & 0 & 0 & \mathbf{0} & x & x \\ 0 & 0 & x & 0 & x & 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & x & x & 0 & 0 & 0 & 0 & 0 & x & x \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & x & \mathbf{0} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x & x & x & 0 & 0 & 0 & 0 & 0 & x \\ 0 & 0 & x & x & x & 0 & 0 & 0 & 0 & x & 0 \end{array}\right].$$

Proceeding with the analogous transformations as in the first subcase, until the row permutation step, we can obtain

$$B_0 = \left[\begin{array}{cc|cccc|cc|cc|cccc|cc} x & x & x & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \\ 0 & x & x & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & x & x & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x & x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & x & x & x & x & x & x & 0 & 0 \end{array}\right].$$

and

$$B_0JB_0^T = \left[\begin{array}{cccccc|cc|cccc} 0&0&0&0&0&0&0&x&x&0&0\\ 0&0&0&0&0&0&0&0&x&0&0\\ 0&0&0&x&x&0&0&0&0&x&x\\ 0&0&x&0&x&0&0&0&0&x&x\\ 0&0&x&x&0&0&0&0&0&x&x\\ \hline 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0\\ \hline x&0&0&0&0&0&0&0&0&0&0\\ x&x&0&0&0&0&0&0&0&0&0\\ 0&0&x&x&x&0&0&0&0&0&x\\ 0&0&x&x&x&0&0&0&0&x&0 \end{array}\right].$$

To maintain the block B_{23} in upper triangular form and to maintain the condition $r \geq p$ we first perform a permutation to move the 5th row of B_0 to the bottom and then perform another permutation to move row 6 to row 5:

$$B_0 = \left[\begin{array}{cc|cccc|cc|cc|cccc|cc} x&x&x&x&x&x&x&x&x&x&x&x&x&x&0&0\\ 0&x&x&x&x&x&x&x&x&x&x&x&x&x&0&0\\ 0&0&x&x&x&x&x&x&x&x&x&x&x&x&0&0\\ 0&0&x&x&x&x&x&x&x&x&x&x&x&x&0&0\\ \hline 0&0&0&0&0&0&x&x&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&0\\ \hline 0&0&0&0&0&0&x&x&x&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&x&x&x&x&0&0&0&0&0&0\\ 0&0&x&x&x&x&x&x&x&x&x&x&x&x&0&0\\ 0&0&x&x&x&x&x&x&x&x&x&x&x&x&0&0\\ 0&0&x&x&x&x&x&x&x&x&x&x&x&x&0&0 \end{array}\right],$$

and

$$B_0JB_0^T = \left[\begin{array}{cccc|cc|ccccc} 0&0&0&0&0&0&x&x&0&0&0\\ 0&0&0&0&0&0&0&x&0&0&0\\ 0&0&0&x&0&0&0&0&x&x&x\\ 0&0&x&0&0&0&0&0&x&x&x\\ \hline 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0\\ \hline x&0&0&0&0&0&0&0&0&0&0\\ x&x&0&0&0&0&0&0&0&0&0\\ 0&0&x&x&0&0&0&0&0&x&x\\ 0&0&x&x&0&0&0&0&x&0&x\\ 0&0&x&x&0&0&0&0&x&x&0 \end{array}\right].$$

Now B_0 and $B_0JB_0^T$ have the forms (4.5) and (4.6), respectively, but $p := p-1$ and $q := q+1$.

Because B_0 is of full row rank, the submatrix consisting of the third and fourth block rows in (4.5) must be of full row rank. Then both B_{23} and the $(1,1)$ block of B_{34} (in lower triangular form) must be nonsingular. Hence during the reductions no diagonal element in B_{34} will be zero, and for deflation we need only to check the diagonal elements of B_{11}. In practice if $B_{11}(j,j)$ satisfies

$$|B_{11}(j,j)| < \varepsilon\|B\|,$$

we set it to zero and perform the deflation step described in case b.

Repeating the above reduction process, we will get (4.4).

We now perform a sequence of orthogonal symplectic transformations to transform (4.4) to (4.2). This is illustrated in the case when $p = 2$, $q = 3$, and $m = 6$:

$$B_0 = \left[\begin{array}{ccc|ccc|ccc|ccc} x & x & x & x & x & x & x & x & x & 0 & 0 & 0 \\ 0 & x & x & x & x & x & x & x & x & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & x & x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & x & x & x & x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & x & x & x & 0 & 0 & 0 & 0 \end{array}\right].$$

Perform the symplectic Givens transformations $G_1 \in \mathcal{G}_3^s(1,4)$, $G_2 \in \mathcal{G}_3^s(2,4)$ on the columns of B_0 to annihilate $B_0(6:7,4)$:

$$B_0 = \left[\begin{array}{ccc|ccc|ccc|ccc} x & x & x & x & x & x & x & x & x & x & 0 & 0 \\ 0 & x & x & x & x & x & x & x & x & x & 0 & 0 \\ \hline 0 & 0 & 0 & x & x & x & x & x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & x & x & x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x & x & x & x & 0 & 0 & 0 & 0 \end{array}\right].$$

In the same way we can annihilate $B_0(6:7,5)$ and $B_0(6:7,6)$:

$$B_0 = \left[\begin{array}{ccc|ccc|ccc|ccc} x & x & x & x & x & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x & x & x & x & x \\ \hline 0 & 0 & 0 & x & x & x & x & x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x & x & x & x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x & x & x & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x & 0 & 0 & 0 & 0 \end{array}\right].$$

Finally perform a symplectic permutation $P \in \mathcal{P}_1^s$ to move columns 3 and 9 to columns 6 and 12, respectively. We have the form (4.2),

$$B_0 = \left[\begin{array}{ccc|ccc|ccc|ccc} x & x & x & x & x & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x & x & x & x & x \\ \hline 0 & 0 & x & x & x & 0 & x & x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & 0 & x & x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x & 0 & x & x & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x & 0 & 0 & 0 & 0 \end{array}\right].$$

5. Error analysis. We only give an error analysis about the eigenvalues. We will provide the first order perturbation bound for a simple nonzero eigenvalue of JB^TB or B^TJB. We will then use the perturbation bound to give the relative error bound for the computed eigenvalues.

5.1. Perturbation about eigenvalues. All nonzero eigenvalues of BJB^T and JBB^T are purely imaginary and they are in conjugate pairs. For real perturbations the perturbation results for both eigenvalues in a conjugate pair are the same. For this reason in the following we consider only the eigenvalues $i\lambda$ with $\lambda > 0$.

Suppose that $i\lambda$ is a simple nonzero eigenvalue of BJB^T and x is a corresponding unit norm eigenvector. Define another unit norm vector

$$y = \frac{JB^T x}{\beta}$$

with $\beta = \|JB^T x\|$. Premultiplying the equation by $JB^T B$, we have

$$JB^T By = i\lambda y.$$

Hence y is a unit norm eigenvector of $JB^T B$ corresponding to $i\lambda$. By using the conjugate transpose of the above equation we have

$$(Jy)^*(JB^T B) = i\lambda (Jy)^*.$$

So Jy is a unit norm left-eigenvector of $JB^T B$. The relation between x, y is summarized as follows:

$$By = i\alpha x, \qquad JB^T x = \beta y, \tag{5.1}$$

where $\alpha = \frac{\lambda}{\beta}$. Taking the conjugate transpose of the second equation in (5.1) and postmultiplying it by Jy,

$$\beta y^* Jy = x^* By.$$

Premultiplying the first equation in (5.1) by x^*,

$$x^* By = i\alpha.$$

The reciprocal of the condition number of $i\lambda$ corresponding to the matrix $JB^T B$ is $\kappa = |(Jy)^* y| = |y^* Jy|$. Combining the above two equations,

$$\kappa = \frac{\alpha}{\beta}. \tag{5.2}$$

Since $\kappa \le 1$ we have $\alpha \le \beta$. Because $\lambda = \alpha\beta$ and $\beta = \|JB^T x\| \le \|B\|$, we have

$$\frac{\lambda}{\|B\|} < \alpha \le \sqrt{\lambda} \le \beta \le \|B\|. \tag{5.3}$$

The first order perturbation bound is given in the following lemma.

LEMMA 5.1. *Suppose that $i\lambda$ ($\lambda > 0$) is a simple eigenvalue of BJB^T and $JB^T B$, and x, y are the corresponding unit norm eigenvectors with respect to BJB^T and $JB^T B$, respectively, satisfying* (5.1). *Let E be a real perturbation matrix and let $\hat{B} = B + E$. When $\|E\|$ is sufficiently small both matrices $\hat{B}J\hat{B}^T$ and $J\hat{B}^T\hat{B}$ have a purely imaginary eigenvalue $i\hat{\lambda}$ such that*

$$\left|\frac{i\hat{\lambda} - i\lambda}{i\lambda}\right| = \left|\frac{2\mathrm{Im}\,(y^* Ex)}{\alpha}\right| + O(\|E\|^2) \le \frac{2\|E\|}{\alpha} + O(\|E\|^2).$$

Proof. The proof follows from the result in [4] for a formal matrix product. ∎

5.2. Error analysis. Again we consider only the case that BJB^T is nonsingular. The general case can be analyzed in the same way. Because of rounding error, the algorithm in section 2 actually computes a block upper triangular matrix R satisfying

$$R = \begin{bmatrix} R_1 & R_2 \\ 0 & R_3 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & R_{14} \\ 0 & 0 & R_{23} & 0 \end{bmatrix} = Q^T(B+E)U,$$

where Q is orthogonal, U is orthogonal symplectic, and E is an error matrix satisfying $\|E\| \le c\varepsilon\|B\|$ for some constant c. Suppose that $i\lambda$ ($\lambda > 0$) is a simple eigenvalue of BJB^T and JB^TB with unit norm eigenvectors x, y satisfying (5.1). When $\|E\|$ is sufficiently small by Lemma 5.1 there is an eigenvalue $i\hat{\lambda}$ of RJR^T and JR^TR such that

$$\left|\frac{i\hat{\lambda} - i\lambda}{i\lambda}\right| = \frac{2|\mathrm{Im}\,(y^*Ex)|}{\alpha} + O(\|E\|^2) \le 2c\varepsilon\frac{\|B\|}{\alpha} + O(\varepsilon^2). \tag{5.4}$$

However, the eigenvalues computed by the algorithm are $\pm i\delta_1, \ldots, \pm i\delta_p$, where $\delta_1, \ldots, \delta_p$ are the diagonal elements of $R_{11}R_{23}^T$. Because of rounding error the product RJR^T is not exactly in the Schur-like form. By a straightforward analysis it satisfies

$$RJR^T = \begin{bmatrix} 0 & \Delta \\ -\Delta & 0 \end{bmatrix} + \begin{bmatrix} F_{11} & F_{12} \\ -F_{12}^T & 0 \end{bmatrix} =: \Gamma + F, \tag{5.5}$$

where $\Delta = \mathrm{diag}(\delta_1, \ldots, \delta_p)$, F_{12} is strictly upper triangular, $F_{11} = -F_{11}^T$, and $\|F\| \le d\varepsilon\|B\|^2$ for some constant d. So the computed eigenvalues are the exact ones of Γ and $i\hat{\lambda}$ in (5.4) is an eigenvalue of $\Gamma + F$. When $\|F\|$ is sufficiently small and we apply the perturbation result [16, sect. 4.2.2], [11, sect. 7.2.2] to $\Gamma + F$ for $i\hat{\lambda}$ there exists a corresponding eigenvalue of Γ, say $i\delta_k$, such that

$$|i\hat{\lambda} - i\delta_k| = |z^*Fz| + O(\|F\|^2),$$

where $z = \frac{\sqrt{2}}{2}(e_k + ie_{p+k})$ is the unit norm eigenvector of $i\delta_k$ (which is obvious from the structure of Γ). Because F_{11} is real skew-symmetric and F_{12} is strictly upper triangular,

$$z^*Fz = \frac{1}{2}(e_k^*F_{11}e_k + 2ie_k^*F_{12}e_k) = 0.$$

Hence $|i\hat{\lambda} - i\delta_k| = O(\varepsilon^2)$. Combining it with (5.4) we have the error bound for $i\delta_k$,

$$\left|\frac{i\delta_k - i\lambda}{i\lambda}\right| = \frac{2|\mathrm{Im}\,(y^*Ex)|}{\alpha} + O(\varepsilon^2) \le 2c\varepsilon\frac{\|B\|}{\alpha} + O(\varepsilon^2). \tag{5.6}$$

For comparison we also give the error bounds for the eigenvalues computed by the numerically backward stable methods working on the explicit product BJB^T or JB^TB. For both matrices explicitly forming the product will introduce an error matrix of order $\varepsilon\|B\|^2$. During the computations another error matrix will be introduced. Here for both matrices JB^TB and BJB^T we assume that the error matrix is of order $\varepsilon\|B\|^2$. (This is true for matrix JB^TB. But for matrix BJB^T the order is $\varepsilon\|BJB^T\|$, which can be much smaller than $\varepsilon\|B\|^2$.) With standard perturbation analysis [16, sect. 4.2.2], [11, sect. 7.2.2] and by using the equality $\lambda = \alpha\beta$ and (5.2), for the simple eigenvalue $i\lambda$, the methods working on BJB^T give an eigenvalue $i\hat{\lambda}_s$ satisfying

$$\left|\frac{i\hat{\lambda}_s - i\lambda}{i\lambda}\right| \le c_s\varepsilon\frac{\|B\|^2}{\lambda} + O(\varepsilon^2) = \left(c_s\varepsilon\frac{\|B\|}{\alpha}\right)\frac{\|B\|}{\beta} + O(\varepsilon^2) \tag{5.7}$$

for some constant c_s. The methods working on JB^TB give an eigenvalue $i\hat{\lambda}_h$ satisfying

$$\left|\frac{i\hat{\lambda}_h - i\lambda}{i\lambda}\right| \le c_h\varepsilon\frac{\|B\|^2}{\lambda\kappa} + O(\varepsilon^2) = \left(c_h\varepsilon\frac{\|B\|}{\alpha}\right)\frac{\|B\|}{\alpha} + O(\varepsilon^2) \tag{5.8}$$

for some constant c_h. By (5.3),

$$\frac{\|B\|}{\alpha} \ge \frac{\|B\|}{\beta} \ge 1.$$

So in general among three bounds (5.6) is the smallest and (5.8) is the biggest. When α or β is small, $\|B\|/\alpha$ or $\|B\|/\beta$ can be much bigger than 1. Since $\lambda = \alpha\beta$, this means that our method can compute tiny eigenvalues more accurately.

6. Numerical examples. We tested and compared the following numerical methods for computing the eigenvalues of the matrices BJB^T and JB^TB.

SSVD. The SVD-like method presented in this paper;

CSVD. The SVD-like method applied to the matrix L^T, where L is the Cholesky factor computed from the explicitly formed matrix $A := B^TB$;

SQR. QR method (bidiagonal-like reduction plus SVD) for BJB^T;

JAC. Jacobi method [15] for BJB^T;

HAM. Hamiltonian method [2, 3] for JB^TB.

All tests were done on a Dell PC with a Pentium 4 processor. All computations were performed in MATLAB version 6.1 with machine precision $\varepsilon \approx 2.22 \times 10^{-16}$.

Example 6.1.

$$B = Q\begin{bmatrix} T^5 & 0 \\ 0 & T^5 \end{bmatrix},$$

where

$$T = \begin{bmatrix} 2 & 1 & & & \\ 1 & 2 & 1 & & \\ & 1 & 2 & 1 & \\ & & 1 & 2 & 1 \\ & & & 1 & 2 \end{bmatrix},$$

and $Q = 5I_{10} - ee^T$ with $e = [1\ \ldots\ 1]^T$. ($Q/5$ is a Householder matrix.) $\|B\| = 3.62 \times 10^3$, $\|BJB^T\| = \|JB^TB\| = \|B\|^2 = 1.31 \times 10^7$.

This example is supposed to test the numerical behavior when no cancellation occurs in forming the product BJB^T. Note that

$$BJB^T = Q\begin{bmatrix} 0 & T^{10} \\ -T^{10} & 0 \end{bmatrix}Q^T, \qquad JB^TB = 25\begin{bmatrix} 0 & T^{10} \\ -T^{10} & 0 \end{bmatrix}.$$

Both matrices have exact eigenvalues $\pm i25[2\cos(k\pi/12)]^{20}$ $(k = 1, \ldots, 5)$. Since all elements of B are integers, no rounding error is introduced in forming the products BJB^T and JB^TB.

The exact eigenvalues and the relative errors of computed eigenvalues are reported in Table 6.1.

In this example for each eigenvalue $i\lambda$, $\alpha = \beta = \sqrt{\lambda}$ and $\kappa = 1$. From Table 6.1 it is clear that *SSVD* gives eigenvalues with relative errors about $\frac{\|B\|}{\sqrt{\lambda}}$ times smaller

TABLE 6.1
Example 6.1*: Exact eigenvalues and relative errors.*

Eigenvalue	rel_{SSVD}	rel_{CSVD}	rel_{SQR}	rel_{JAC}	rel_{HAM}
$\pm i4.77 \times 10^{-5}$	4.0×10^{-12}	4.7×10^{-7}	2.9×10^{-6}	1.2×10^{-6}	6.0×10^{-6}
$\pm i2.50 \times 10^{1}$	3.8×10^{-15}	4.2×10^{-12}	6.0×10^{-12}	2.9×10^{-12}	4.6×10^{-12}
$\pm i2.56 \times 10^{4}$	2.0×10^{-15}	2.0×10^{-15}	9.2×10^{-15}	3.6×10^{-15}	5.7×10^{-15}
$\pm i1.48 \times 10^{6}$	1.1×10^{-15}	1.4×10^{-15}	1.6×10^{-15}	1.6×10^{-15}	1.7×10^{-15}
$\pm i1.31 \times 10^{7}$	7.1×10^{-16}	0	1.4×10^{-16}	8.5×10^{-16}	5.7×10^{-16}

TABLE 6.2
Example 6.1*: Residuals and errors.*

	SSVD	*CSVD*	*SQR*	*JAC*
err_S	4.6×10^{-13}	4.0×10^{-13}	2.9×10^{-6}	1.2×10^{-6}
res_B	1.3×10^{-15}	–	2.8×10^{-16}	8.9×10^{-16}
res_{JA}	1.6×10^{-15}	1.3×10^{-15}	1.7×10^{-16}	3.2×10^{-16}
res_{SCF}	2.1×10^{-13}	1.9×10^{-13}	3.5×10^{-11}	9.1×10^{-11}

than other methods. *CSVD* is basically the same as other methods. This is because computing the Cholesky factorization already introduced an error of order $O(\varepsilon\|B\|^2)$ to A.

We also computed the following quantities:

$$err_S = \max\{\|SJS^T - J\|, \|S^TJS - J\|\}, \qquad res_B = \frac{\|QDS^{-1} - B\|}{\|B\|},$$
$$res_{JA} = \frac{\|S(JD^TD)S^{-1} - JB^TB\|}{\|JB^TB\|}, \qquad res_{SCF} = \frac{\|JD^TD - S^{-1}(JB^TB)S\|}{\|JD^TD\|},$$

where QDS^{-1} is the SVD-like decomposition of B. These quantities are used to measure the accuracy of the symplectic matrix S, the residual of the SVD-like decomposition of B, the residual of the canonical form of JB^TB, and the accuracy of the eigenvectors, respectively. The matrices S and D are computed as follows. With *SSVD* and *CSVD*, S is computed by using (2.5) and $D = \mathrm{diag}(\Sigma, \Sigma)$. With *SQR* and *JAC*, after obtaining the Schur-like form

$$BJB^T = Q \begin{bmatrix} 0 & \Delta \\ -\Delta & 0 \end{bmatrix} Q^T,$$

we set $D = \mathrm{diag}(\sqrt{\Delta}, \sqrt{\Delta})$. Let $Z := D^{-1}Q^TB$. Then $B = QDZ$ and

$$ZJZ^T = D^{-1}Q^TBJB^TQD^{-1} = D^{-1} \begin{bmatrix} 0 & \Delta \\ -\Delta & 0 \end{bmatrix} D^{-1} = J.$$

So we take Z^{-1} as S. Since Z is symplectic, $Z^{-1} = JZ^TJ^T$. In practice we use the formula $S = JB^TQD^{-1}J^T$ to compute S. The computed results are reported in Table 6.2. Both *SQR* and *JAC* give slightly smaller residuals res_B and res_{JA}. But both *SSVD* and *CSVD* give much smaller err_S, indicating that the matrix S computed by *SSVD* and *CSVD* is more "symplectic."

Example 6.2.

$$B = Q \begin{bmatrix} \Sigma & 0 \\ 0 & \Sigma \end{bmatrix} \left(\begin{bmatrix} X & X \\ 0 & X^{-1} \end{bmatrix} V^T \right),$$

TABLE 6.3
Example 6.2*: Exact eigenvalues and relative errors.*

Eigenvalue	rel_{SSVD}	rel_{CSVD}	rel_{SQR}	rel_{JAC}	rel_{HAM}
$\pm i$	6.9×10^{-15}	1.1×10^{-10}	2.7×10^{-14}	2.8×10^{-14}	5.8×10^{-10}
$\pm 4i$	1.2×10^{-13}	6.6×10^{-9}	8.0×10^{-14}	8.1×10^{-14}	1.5×10^{-8}
$\pm 9i$	5.3×10^{-15}	1.2×10^{-13}	2.0×10^{-15}	9.9×10^{-16}	2.2×10^{-13}
$\pm 16i$	2.8×10^{-14}	6.4×10^{-13}	3.4×10^{-14}	3.4×10^{-14}	4.9×10^{-12}
$\pm 25i$	1.6×10^{-15}	5.5×10^{-12}	1.3×10^{-15}	8.5×10^{-16}	1.5×10^{-10}

TABLE 6.4
Example 6.2*: Relative error bounds.*

Eigenvalue	$2\varepsilon\frac{\|B\|}{\alpha}$	$\varepsilon\frac{\|B\|^2}{\lambda}$	$\varepsilon\frac{\|B\|^2}{\alpha^2}$
$\pm i$	2.2×10^{-11}	1.1×10^{-10}	5.6×10^{-7}
$\pm 4i$	1.1×10^{-12}	2.8×10^{-11}	5.6×10^{-9}
$\pm 9i$	1.3×10^{-13}	1.2×10^{-12}	1.7×10^{-10}
$\pm 16i$	7.8×10^{-13}	6.9×10^{-12}	1.1×10^{-8}
$\pm 25i$	6.3×10^{-12}	4.4×10^{-12}	1.1×10^{-6}

where $\Sigma = \mathrm{diag}(5, 4, 3, 2, 1)$ and $X = \mathrm{diag}(100, 10, 1, 0.1, 0.01)$, Q is a random orthogonal matrix, and V is a random orthogonal symplectic matrix. $\|B\| = 7.07 \times 10^2$, $\|B\|^2 = 5.00 \times 10^5$.

This example is supposed to test the numerical behavior when big cancellation takes place in forming the product BJB^T ($\|BJB^T\| = 25$). The exact eigenvalues and the relative errors of the computed eigenvalues are reported in Table 6.3. For each eigenvalue $i\lambda$ the relative error bounds (5.6)–(5.8) are given in Table 6.4. (Here we set $c = c_s = c_h = 1$.)

Because for the Hamiltonian matrix JB^TB its eigenvalues have relatively big condition numbers, *HAM* gives less accurate eigenvalues. Again, *CSVD* also gives less accurate eigenvalues because of the Cholesky factorization. The other three methods compute the eigenvalues with the same accuracy, as predicted by the error bounds. The residuals of the decompositions and err_S, res_{SCF} are reported in Table 6.5. In this example all these methods basically give the same results.

Example 6.3.

$$B = Q \left[\begin{array}{ccc|ccc} \Sigma & 0 & 0 & 0 & 0 & 0 \\ 0 & I_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Sigma & 0 & 0 \end{array}\right] U^T,$$

where $\Sigma = \mathrm{diag}(10^{-4}, 10^{-2}, 1, 10^2)$, Q is a random orthogonal matrix, and U is a 14×14 random orthogonal symplectic matrix. $\|B\| = 10^2$ and $\|B\|^2 = 10^4$.

This example is supposed to test the numerical behavior when BJB^T has (two) zero eigenvalues. The exact eigenvalues, the absolute errors for zero eigenvalues, and the relative errors for nonzero eigenvalues are reported in Table 6.6.

In this example for zero eigenvalues *SSVD* gives the eigenvalues of order ε, while *SQR*, *JAC*, and *HAM* give answers about $\|B\|$ times bigger than *SSVD*.[4] For nonzero eigenvalues, as in Example 6.1, *SSVD* gives the results with relative errors about $\frac{\|B\|}{\sqrt{\lambda}}$ times smaller than those of the other methods.

[4]The matrix JB^TB actually has two additional 2×2 Jordan blocks corresponding to zero eigenvalues. The corresponding eigenvalues computed by *HAM* are $\pm 8.37 \times 10^{-8} \pm 2.64 \times 10^{-7} i$.

TABLE 6.5
Example 6.2*: Residuals and errors.*

	$SSVD$	$CSVD$	SQR	JAC
err_S	8.8×10^{-12}	3.4×10^{-11}	3.2×10^{-12}	5.5×10^{-13}
res_B	1.2×10^{-15}	–	3.8×10^{-16}	1.6×10^{-15}
res_{JA}	1.3×10^{-15}	1.8×10^{-15}	2.1×10^{-16}	1.0×10^{-15}
res_{SCF}	3.1×10^{-9}	6.5×10^{-10}	3.1×10^{-9}	3.1×10^{-9}

TABLE 6.6
Exact eigenvalues and errors for Example 6.3.

Eigenvalue	rel_{SSVD}	rel_{SQR}	rel_{JAC}	rel_{HAM}
0(double)	1.7×10^{-15}	1.1×10^{-14}	5.7×10^{-14}	1.5×10^{-13}
$\pm i10^{-8}$	1.9×10^{-11}	8.9×10^{-6}	1.3×10^{-5}	5.9×10^{-6}
$\pm i10^{-4}$	5.7×10^{-13}	1.7×10^{-9}	4.1×10^{-11}	7.5×10^{-10}
$\pm i$	1.3×10^{-15}	1.1×10^{-13}	1.1×10^{-14}	2.1×10^{-13}
$\pm i10^{4}$	1.8×10^{-16}	1.8×10^{-16}	1.3×10^{-15}	3.6×10^{-16}

In this example we did not test *CSVD*. Because in this case it is more complicated to compute the matrix S by *SQR* and *JAC*, we did not compare the residuals and err_S, res_{SCF}.

7. Conclusion. We have developed a numerical method to compute the SVD-like decomposition of a real matrix B. The method can be simply applied to compute the eigenvalues and canonical forms of the skew-symmetric matrix BJB^T and the Hamiltonian matrix JB^TB. Unlike other numerical methods this method works only on the factor B. In this way the eigenvalues (particularly the small eigenvalues) of BJB^T and JB^TB can be computed more accurately. This has been demonstrated by the error bound and several numerical examples. The numerical examples also show that the symplectic matrix S computed by the proposed method is more accurate.

Acknowledgment. The author gratefully acknowledges the anonymous reviewers for their valuable comments and suggestions on the first version of this paper.

REFERENCES

[1] P. BENNER, R. BYERS, V. MEHRMANN, AND H. XU, *Numerical computation of deflating subspaces of skew-Hamiltonian/Hamiltonian pencils*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 165–190.
[2] P. BENNER, V. MEHRMANN, AND H. XU, *A new method for computing the stable invariant subspace of a real Hamiltonian matrix*, J. Comput. Appl. Math., 86 (1997), pp. 17–43.
[3] P. BENNER, V. MEHRMANN, AND H. XU, *A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils*, Numer. Math., 78 (1998), pp. 329–358.
[4] P. BENNER, V. MEHRMANN, AND H. XU, *Perturbation analysis for the eigenvalue problem of a formal product of matrices*, BIT, 42 (2002), pp. 1–43.
[5] A. BOJANCZYK, G.H. GOLUB, AND P. VAN DOOREN, *The periodic Schur decomposition. Algorithms and applications*, in Advanced Signal Processing Algorithms, Architectures, and Implementations III, Proc. SPIE 1770, SPIE, Bellingham, WA, 1992, pp. 31–42.
[6] P.A. BUSINGER AND G.H. GOLUB, *Linear least squares solutions by Householder transformations*, Numer. Math., 7 (1965), pp. 269–276.
[7] T. CHAN, *Rank revealing QR factorizations*, Linear Algebra Appl., 88/89 (1987), pp. 67–82.
[8] R.J. DUFFIN, *The Rayleigh-Ritz method for dissipative or gyroscopic systems*, Quart. Appl. Math., 18 (1960), pp. 215–221.
[9] G. GOLUB AND W. KAHAN, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM J. Numer. Anal., 2 (1965), pp. 205–224.

[10] G. Golub, K. Sølna, and P. Van Dooren, *Computing the SVD of a general matrix product/quotient*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 1–19.
[11] G. Golub and C.F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[12] 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.
[13] P. Lancaster, *Lambda-Matrices and Vibrating Systems*, Pergamon Press, Oxford, UK, 1966.
[14] C.B. Moler and G.W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, SIAM J. Numer. Anal., 10 (1973), pp. 241–256.
[15] M.H.C. Paardekooper, *An eigenvalue algorithm for skew-symmetric matrices*, Numer. Math., 17 (1971), pp. 189–202.
[16] G.W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, New York, 1990.
[17] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.
[18] H. Xu, *An SVD-like matrix decomposition and its applications*, Linear Algebra Appl., 368 (2003), pp. 1–24.
[19] V.A. Yakubovich and V.M. Starzhinskii, *Linear Differential Equations with Periodic Coefficients,* Vols. 1 and 2, Halstead, New York, Toronto, 1975.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 4, pp. 1083–1099

BLOCK-TOEPLITZ/HANKEL STRUCTURED TOTAL LEAST SQUARES*

IVAN MARKOVSKY†, SABINE VAN HUFFEL†, AND RIK PINTELON‡

Abstract. A structured total least squares problem is considered in which the extended data matrix is partitioned into blocks and each of the blocks is block-Toeplitz/Hankel structured, unstructured, or exact. An equivalent optimization problem is derived and its properties are established. The special structure of the equivalent problem enables us to improve the computational efficiency of the numerical solution methods. By exploiting the structure, the computational complexity of the algorithms (local optimization methods) per iteration is linear in the sample size. Application of the method for system identification and for model reduction is illustrated by simulation examples.

Key words. parameter estimation, total least squares, structured total least squares, system identification, model reduction

AMS subject classifications. 15A06, 62J12, 37M10

DOI. 10.1137/S0895479803434902

1. Introduction.

The *total least squares* (TLS) problem

$$\min_{\Delta A, \Delta B, X} \left\| \begin{bmatrix} \Delta A & \Delta B \end{bmatrix} \right\|_{\mathrm{F}}^2 \quad \text{subject to} \quad (A - \Delta A)X = B - \Delta B, \tag{1.1}$$

where $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{m \times d}$, $C := [A\ B]$ is the data matrix, and $X \in \mathbb{R}^{n \times d}$ is the parameter of interest, proved to be a useful parameter estimation technique. It became especially popular since the early eighties due to the development [8] of reliable solution methods based on singular value decomposition. The same technique is known in the system identification literature as the Koopmans–Levin method [12] and in the statistical literature as orthogonal regression [7]. For a comprehensive introduction to the theory, algorithms, and applications of the TLS method, see [25].

With the increased interest in the TLS technique, more and more researchers started to apply it in various applications. In some cases, however, important assumptions of the method are not satisfied, which resulted in the development of appropriate extensions of the original TLS method. We mention the *mixed LS-TLS method*

*Received by the editors September 15, 2003; accepted for publication (in revised form) by L. Eldén June 30, 2004; published electronically May 6, 2005. This research was supported by Research Council K. U. Leuven through grants GOA-Mefisto 666, IDO/99/003, and IDO/02/009 (predictive computer models for medical classification problems using patient data and expert knowledge) and several Ph.D./postdoctorate and fellow grants; the Flemish Government, FWO, through Ph.D./postdoctorate grants, projects, and grants G.0200.00 (damage detection in composites by optical fibers), G.0078.01 (structured matrices), G.0407.02 (support vector machines), G.0269.02 (magnetic resonance spectroscopic imaging), and G.0270.02 (nonlinear Lp approximation); research communities (ICCoS, ANMMM); the AWI under the Bil. Int. Collaboration Hungary/Poland; the IWT through Ph.D. grants; the Belgian Federal Government, DWTC (grants IUAP IV-02 (1996–2001) and IUAP V-22 (2002–2006): Dynamical Systems and Control: Computation, Identification & Modelling); the EU through NICONET, INTERPRET, PDT-COIL, MRS/MRI signal processing (TMR); and contract research/agreements (Data4s, IPCOS).

http://www.siam.org/journals/simax/26-4/43490.html

†ESAT-SCD (SISTA), Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium (Ivan.Markovsky@esat.kuleuven.ac.be, Sabine.VanHuffel@esat.kuleuven.ac.be). The first author was supported by a K.U. Leuven doctoral scholarship.

‡Department ELEC, Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussels, Belgium (Rik.Pintelon@vub.ac.be).

[25, sect. 3.5], where some of the columns of A are exact (noise free), and the so-called *generalized total least squares method* [24], where the cost function of TLS problem (1.1) is generalized to $\| \begin{bmatrix} \Delta A & \Delta B \end{bmatrix} V \|_{\mathrm{F}}^2$, with $V \geq 0$. The latest developments in the field are collected in the proceedings books [22, 23].

In the early nineties, a powerful generalization of the TLS method was put forward [1, 4, 20]. The so-called *structured total least squares* (STLS) problem,

$$\min_{\Delta A, \Delta B, X} \left\| \begin{bmatrix} \Delta A & \Delta B \end{bmatrix} \right\|_{\mathrm{F}}^2 \quad \text{subject to} \quad (A - \Delta A)X = B - \Delta B \quad \text{and} \quad \begin{bmatrix} \Delta A & \Delta B \end{bmatrix} \text{ has the same structure as } \begin{bmatrix} A & B \end{bmatrix},$$

defined in the same way as the TLS problem (1.1) but with the additional structure constraint, includes as special cases many of the presently known TLS variations. The structure occurs naturally in, e.g., applications dealing with discrete-time dynamic phenomena [6], where the Hankel and Toeplitz matrices are fundamental.

Although the STLS problem is very general, it is still not widely accepted due to the lack of reliable solution methods for its computation, the main difficulty being that, as an optimization problem, it is nonconvex and there is no guarantee that a global minimum point will be found. Still, under certain conditions [10] for highly overdetermined systems ($m \ll nd$) the solution of the problem is unique and the main difficulty—the presence of multiple local minima—tends to disappear for large sample sizes (i.e., for $m \to \infty$). In addition, due to the consistency results of [10], such an assumption guarantees accurate estimation and makes the problem meaningful from a statistical point of view.

However, the currently used numerical algorithms for solving the STLS problem can hardly deal with large sample sizes. The original methods of [1, 4, 20] have computational costs that increase quadratically or even cubically as a function of m. In [11, 17], methods with computational cost linear in m are developed using the generalized Schur algorithm. These methods, however, are developed for a particular structure of the data matrix C (in [11] C is Hankel, and in [17] $A \in \mathbb{R}^{m \times n}$ is Toeplitz and $B \in \mathbb{R}^{m \times 1}$ is unstructured) and modifications for other structures are nontrivial.

In [13], based on the insight from [10], we have proposed a new approach with computational cost linear in m and dealing with a flexible structure specification. The data matrix C can be partitioned into blocks $C = \begin{bmatrix} C^{(1)} & \cdots & C^{(q)} \end{bmatrix}$, where each of the blocks $C^{(l)}$, for $l = 1, \ldots, q$, is Hankel, Toeplitz, unstructured, or exact.

In this paper, we consider an extension of the results of [13] to the case of block-Hankel and block-Toeplitz structured matrices. Thus the data matrix is now a block matrix, of which the blocks are themselves structured with one of the four possible structures: block-Hankel, block-Toeplitz, unstructured, or exact. The need for such an extension comes from applications dealing with multi-input and/or multi-output dynamical systems. The proposed algorithms are implemented in C (see [15]), and the software is available.

Standard notation used in the paper is as follows: $\mathbb{R}$ for the set of the real numbers, $\mathbb{N}$ for the set of the natural numbers, $\| \cdot \|$ for the Euclidean norm, and $\| \cdot \|_{\mathrm{F}}$ for the Frobenius norm. The operator that vectorizes columnwise a matrix is denoted by $\mathrm{vec}(\cdot)$, the expectation operator by $\mathbf{E}$, and the covariance matrix of a random vector by $\mathrm{cov}(\cdot)$. The pseudoinverse of a matrix A is denoted by $A^{\dagger}$.

2. The STLS problem. In this section, we define the STLS problem, considered in the paper, and derive an equivalent optimization problem. Consider a function $\mathcal{S} : \mathbb{R}^{n_p} \to \mathbb{R}^{m \times (n+d)}$ that defines the structure of the data as follows: a matrix $C \in \mathbb{R}^{m \times (n+d)}$ is said to have the structure defined by $\mathcal{S}$ if there exists a $p \in \mathbb{R}^{n_p}$,

such that $C = \mathcal{S}(p)$. The vector p is called a *parameter vector* for the structured matrix C.

PROBLEM 2.1 (STLS problem). *Given a data vector $p \in \mathbb{R}^{n_p}$ and a structure specification $\mathcal{S} : \mathbb{R}^{n_p} \to \mathbb{R}^{m\times(n+d)}$, solve the optimization problem*

$$\min_{X,\Delta p} \|\Delta p\|^2 \quad \textit{subject to} \quad \mathcal{S}(p-\Delta p)\begin{bmatrix} X \\ -I_d \end{bmatrix} = 0. \tag{2.1}$$

The interpretation of (2.1) is the following: Find the smallest correction Δp, measured in 2-norm, that makes the structured matrix $\mathcal{S}(p-\Delta p)$ rank deficient with rank at most n. Define

$$X_{\text{ext}} := \begin{bmatrix} X \\ -I_d \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} A & B \end{bmatrix} := C := \mathcal{S}(p), \text{ where } A \in \mathbb{R}^{m\times n} \text{ and } B \in \mathbb{R}^{m\times d}.$$

$CX_{\text{ext}} = 0$ is shorthand notation for the structured system of equations $AX = B$.

The STLS problem is said to be *affine structured* if the function $\mathcal{S}$ is affine, i.e.,

$$\mathcal{S}(p) = S_0 + \sum_{i=1}^{n_p} S_i p_i \quad \text{for all } p \in \mathbb{R}^{n_p} \text{ and for some } S_i,\ i = 1,\dots,n_p. \tag{2.2}$$

In an affine STLS problem, the constraint $\mathcal{S}(p-\Delta p)X_{\text{ext}} = 0$ becomes bilinear in the decision variables X and Δp.

LEMMA 2.2. *Let $\mathcal{S} : \mathbb{R}^{n_p} \to \mathbb{R}^{m\times(n+d)}$ be an affine function. Then*

$$\mathcal{S}(p-\Delta p)X_{ext} = 0 \iff G(X)\Delta p = r(X),$$

where

$$G(X) := \begin{bmatrix} \operatorname{vec}\big((S_1 X_{ext})^\top\big) & \cdots & \operatorname{vec}\big((S_{n_p} X_{ext})^\top\big) \end{bmatrix} \in \mathbb{R}^{md\times n_p} \tag{2.3}$$

and

$$r(X) := \operatorname{vec}\big(\big(\mathcal{S}(p)X_{ext}\big)^\top\big) \in \mathbb{R}^{md}.$$

Proof.

$$\begin{aligned} \mathcal{S}(p-\Delta p)X_{\text{ext}} = 0 &\iff \sum_{i=1}^{n_p} S_i \Delta p_i X_{\text{ext}} = \mathcal{S}(p)X_{\text{ext}} \\ &\iff \sum_{i=1}^{n_p} \operatorname{vec}\big((S_i X_{\text{ext}})^\top\big)\Delta p_i = \operatorname{vec}\big(\big(\mathcal{S}(p)X_{\text{ext}}\big)^\top\big) \\ &\iff G(X)\Delta p = r(X). \qquad \square \end{aligned}$$

Using Lemma 2.2, we rewrite the affine STLS problem as follows:

$$\min_X \Big(\min_{\Delta p} \|\Delta p\|^2 \quad \text{subject to} \quad G(X)\Delta p = r(X) \Big). \tag{2.4}$$

The inner minimization problem has an analytic solution, which allows us to derive an equivalent optimization problem.

THEOREM 2.3 (equivalent optimization problem for affine STLS). *Assuming that* $n_p \geq md$, *the affine STLS problem* (2.4) *is equivalent to*

$$\min_X f_0(X), \text{ where } f_0(X) := r^\top(X)\Gamma^\dagger(X)r(X) \quad \text{and} \quad \Gamma(X) := G(X)G^\top(X). \tag{2.5}$$

Proof. Under the assumption $n_p \geq md$, the inner minimization problem of (2.4) is a least norm problem. Its minimum point (as a function of X) is

$$\Delta p_{\min}(X) = G^\top(X)\big(G(X)G^\top(X)\big)^\dagger r(X),$$

so that

$$f_0(X) = \Delta p_{\min}^\top(X)\Delta p_{\min}(X) = r^\top(X)\big(G(X)G^\top(X)\big)^\dagger r(X) = r^\top(X)\Gamma^\dagger(X)r(X). \qquad \square$$

The significance of Theorem 2.3 is that the constraint and the decision variable Δp in problem (2.4) are eliminated. Note that typically the number of elements nd in X is much smaller than the number of elements n_p in the correction Δp. Thus the reduction in the complexity is significant.

The equivalent optimization problem (2.5) is a nonlinear least squares problem, so that classical optimization methods can be used for its solution. The optimization methods require a cost function and first derivative evaluation. In order to evaluate the cost function f_0 for a given value of the argument X, we need to form the weight matrix $\Gamma(X)$ and to solve the system of equations $\Gamma(X)y(X) = r(X)$. This straightforward implementation requires $O(m^3)$ floating point operation (flops). For large m (the applications that we aim at) this computational complexity becomes prohibitive.

It turns out, however, that for a special case of affine structures $\mathcal{S}$, the weight matrix $\Gamma(X)$ is nonsingular and has a block-Toeplitz and block-banded structure, which can be exploited for efficient cost function and first derivative evaluations. The set of structures of $\mathcal{S}$, for which we establish the special properties of $\Gamma(X)$, is

$$\begin{aligned} \mathcal{S}(p) = \begin{bmatrix} C^{(1)} & \cdots & C^{(q)} \end{bmatrix} &\text{ for all } p \in \mathbb{R}^{n_p}, \text{ where } C^{(l)}, \text{ for } l = 1, \ldots, q, \text{ is} \\ &\text{block-Toeplitz, block-Hankel, exact, or unstructured} \\ &\text{and all block-Toeplitz/Hankel structured blocks } C^{(l)} \\ &\text{have equal row dimension } K \text{ of the blocks.} \end{aligned} \tag{2.6}$$

Assumption (2.6) says that $\mathcal{S}(p)$ is composed of blocks, each of which is block-Toeplitz, block-Hankel, exact, or unstructured. A block $C^{(l)}$ that is exact is not modified in the solution $\hat{C} := \mathcal{S}(p - \Delta p)$, i.e., $\hat{C}^{(l)} = C^{(l)}$. Assumption 2.6 is the essential structural assumption that we impose on problem (2.1). As shown in section 6, it is fairly general and covers many applications.

Example 1. Consider the block-Toeplitz matrix

$$C = \left[\begin{array}{c|c|c} \begin{matrix} 5 \\ 6 \end{matrix} & \begin{matrix} 3 \\ 4 \end{matrix} & \begin{matrix} 1 \\ 2 \end{matrix} \\ \hline \begin{matrix} 7 \\ 8 \end{matrix} & \begin{matrix} 5 \\ 6 \end{matrix} & \begin{matrix} 3 \\ 4 \end{matrix} \\ \hline \begin{matrix} 9 \\ 10 \end{matrix} & \begin{matrix} 7 \\ 8 \end{matrix} & \begin{matrix} 5 \\ 6 \end{matrix} \end{array}\right]$$

with row dimension of the block $K = 2$. Next we specify the matrices S_i that define via (2.2) an affine function $\mathcal{S}$, such that $C = \mathcal{S}(p)$ for a certain parameter vector p. Let == be an elementwise comparison operator. Acting on matrices of the same size, it gives as a result a matrix with the same size as the arguments, of which the (i, j)th element is 1 if the corresponding elements of the arguments are equal, and 0 otherwise. (Think of MATLAB's == operator.) Let E be the 6×3 matrix with all elements equal to 1 and define $S_0 := 0_{6\times 3}$ and $S_i := (C == iE)$ for $i = 1, \dots, 10$. We have

$$C = \sum_{i=1}^{10} S_i i = S_0 + \sum_{i=1}^{10} S_i p_i =: \mathcal{S}(p), \qquad \text{with } p = \begin{bmatrix} 1 & 2 & \cdots & 10 \end{bmatrix}^\top.$$

The matrix C considered in the example is special; it allowed us to easily write down a corresponding affine function $\mathcal{S}$. Clearly with the constructed $\mathcal{S}$, *any* 6×3 block-Toeplitz matrix C with row dimension of the block $K = 2$ can be written as $C = \mathcal{S}(p)$ for certain $p \in \mathbb{R}^{10}$.

We will use the notation $\mathbf{n}_l$ for the number of *block* columns of the block $C^{(l)}$. For unstructured and exact blocks, $\mathbf{n}_l := 1$.

3. Properties of the weight matrix Γ. For the evaluation of the cost function f_0 of the equivalent optimization problem (2.5), we have to solve the system of equations $\Gamma(X)y(X) = r(X)$, where $\Gamma(X) \in \mathbb{R}^{md\times n_p}$ with both m and n_p large. In this section, we investigate the structure of the matrix $\Gamma(X)$. Occasionally we drop the explicit dependence of r and Γ on X.

THEOREM 3.1 (structure of the weight matrix Γ). *Consider the equivalent optimization problem* (2.5) *from Theorem* 2.3. *If, in addition to the assumptions of Theorem* 2.3, *the structure $\mathcal{S}$ is such that* (2.6) *holds, then the weight matrix $\Gamma(X)$ has the block-banded Toeplitz structure*

$$\Gamma(X) = \begin{bmatrix} \Gamma_0 & \Gamma_1^\top & \cdots & \Gamma_s^\top & & \mathbf{0} \\ \Gamma_1 & \ddots & \ddots & \ddots & \ddots & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \Gamma_s^\top \\ \Gamma_s & \ddots & \ddots & \ddots & \ddots & \vdots \\ & \ddots & \ddots & \ddots & \ddots & \Gamma_1^\top \\ \mathbf{0} & & \Gamma_s & \cdots & \Gamma_1 & \Gamma_0 \end{bmatrix} \in \mathbb{R}^{md\times md}, \tag{3.1}$$

where $\Gamma_k \in \mathbb{R}^{dK\times dK}$, for $k = 0, 1, \dots, s$, and $s = \max_{l=1,\dots,q}(\mathbf{n}_l - 1)$, where $\mathbf{n}_l$ is the number of block columns in the block $C^{(l)}$ of the data matrix $\mathcal{S}(p)$.

The proof is developed in a series of lemmas. First we reduce the original problem with multiple blocks $C^{(l)}$ (see (2.6)) to three independent problems—one for the unstructured case, one for the block-Hankel case, and one for the block-Toeplitz case.

LEMMA 3.2. *Consider a structure specification of the form*

$$\mathcal{S}(p) = \begin{bmatrix} \mathcal{S}^{(1)}(p^{(1)}) & \cdots & \mathcal{S}^{(q)}(p^{(q)}) \end{bmatrix}, \qquad p^{(l)} \in \mathbb{R}^{n_p^{(l)}}, \quad \sum_{l=1}^{q} n_p^{(l)} =: n_p,$$

where $p^\top =: \begin{bmatrix} p^{(1)\top} & \cdots & p^{(q)\top} \end{bmatrix}$ and $\mathcal{S}(p^{(l)}) := S_0^{(l)} + \sum_{i=1}^{n_p^{(l)}} S_i^{(l)} p_i^{(l)}$ for all $p^{(l)} \in \mathbb{R}^{n_p^{(l)}}$,

$l = 1, \ldots, q$. *Then*

$$\Gamma(X) = \sum_{l=1}^{q} \Gamma^{(l)}(X), \tag{3.2}$$

where $\Gamma^{(l)} := G^{(l)} G^{(l)\top}$, $G^{(l)} := [\operatorname{vec}((S_1^{(l)} X_{ext}^{(l)})^\top) \quad \cdots \quad \operatorname{vec}((S_{n_p^{(l)}}^{(l)} X_{ext}^{(l)})^\top)]$, *and*

$$X_{ext} =: \begin{bmatrix} X_{ext}^{(1)} \\ \vdots \\ X_{ext}^{(q)} \end{bmatrix}, \quad \textit{with} \quad X_{ext}^{(l)} \in \mathbb{R}^{n_l \times d}, \quad n_l := \operatorname{coldim}(C^{(l)}), \quad \sum_{l=1}^{q} n_l = n + d.$$

Proof. The result is a refinement of Lemma 2.2. Let $\Delta p^\top =: \begin{bmatrix} \Delta p^{(1)\top} & \cdots & \Delta p^{(q)\top} \end{bmatrix}$, where $\Delta p^{(l)} \in \mathbb{R}^{n_p^{(l)}}$ for $l = 1, \ldots, q$. We have

$$\begin{aligned} \mathcal{S}(p - \Delta p) X_{\text{ext}} = 0 &\iff \sum_{l=1}^{q} \mathcal{S}^{(l)}(p^{(l)} - \Delta p^{(l)}) X_{\text{ext}}^{(l)} = 0 \\ &\iff \sum_{l=1}^{q} \sum_{i=1}^{n_p} S_i^{(l)} \Delta p_i^{(l)} X_{\text{ext}}^{(l)} = \mathcal{S}(p) X_{\text{ext}} \\ &\iff \sum_{l=1}^{q} G^{(l)} \Delta p^{(l)} = r(X) \\ &\iff \underbrace{\begin{bmatrix} G^{(1)} & \cdots & G^{(q)} \end{bmatrix}}_{G(X)} \Delta p = r(X), \end{aligned}$$

so that $\Gamma = GG^\top = \sum_{l=1}^{q} G^{(l)} G^{(l)\top} = \sum_{l=1}^{q} \Gamma^{(l)}$. ∎

Next we establish the structure of Γ for an STLS problem with an unstructured data matrix.

LEMMA 3.3. *Let*

$$\mathcal{S}(p) := \begin{bmatrix} p_1 & p_2 & \cdots & p_{n+d} \\ p_{n+d+1} & p_{n+d+2} & \cdots & p_{2(n+d)} \\ \vdots & \vdots & & \vdots \\ p_{(m-1)(n+d)+1} & p_{(m-1)(n+d)+2} & \cdots & p_{m(n+d)} \end{bmatrix} \in \mathbb{R}^{m \times (n+d)};$$

then

$$\Gamma = I_m \otimes (X_{ext}^\top X_{ext}); \tag{3.3}$$

i.e., the matrix Γ *has the structure* (3.1) *with* $s = 0$ *and* $\Gamma_0 = I_K \otimes (X_{ext}^\top X_{ext})$.

Proof. We have

$$\begin{aligned} \mathcal{S}(p - \Delta p) X_{\text{ext}} = 0 &\iff \operatorname{vec}\big(X_{\text{ext}}^\top \mathcal{S}^\top(\Delta p)\big) = \operatorname{vec}\big(\big(\mathcal{S}(p) X_{\text{ext}}\big)^\top\big) \\ &\iff \underbrace{(I_m \otimes X_{\text{ext}}^\top)}_{G(X)} \underbrace{\operatorname{vec}\big(\mathcal{S}^\top(\Delta p)\big)}_{\Delta p} = r(X). \end{aligned}$$

Therefore, $\Gamma = GG^\top = (I_m \otimes X_{\text{ext}}^\top)(I_m \otimes X_{\text{ext}}^\top)^\top = I_m \otimes (X_{\text{ext}}^\top X_{\text{ext}})$. ∎

Next we establish the structure of Γ for an STLS problem with a block-Hankel data matrix.

LEMMA 3.4. *Let*

$$\mathcal{S}(p) := \begin{bmatrix} C_1 & C_2 & \cdots & C_{\mathbf{n}} \\ C_2 & C_3 & \cdots & C_{\mathbf{n}+1} \\ \vdots & \vdots & & \vdots \\ C_{\mathbf{m}} & C_{\mathbf{m}+1} & \cdots & C_{\mathbf{m}+\mathbf{n}-1} \end{bmatrix} \in \mathbb{R}^{m\times(n+d)}, \qquad \begin{aligned} \mathbf{n} &:= \frac{n+d}{L}, \\ \mathbf{m} &:= \frac{m}{K}, \end{aligned}$$

where C_i are $K \times L$ unstructured blocks, parameterized by $p^{(i)} \in \mathbb{R}^{KL}$ as follows:

$$C_i := \begin{bmatrix} p_1^{(i)} & p_2^{(i)} & \cdots & p_L^{(i)} \\ p_{L+1}^{(i)} & p_{L+2}^{(i)} & \cdots & p_{2L}^{(i)} \\ \vdots & \vdots & & \vdots \\ p_{(K-1)L+1}^{(i)} & p_{(K-1)L+2}^{(i)} & \cdots & p_{KL}^{(i)} \end{bmatrix} \in \mathbb{R}^{K\times L}.$$

Define a partitioning of X_{ext} as follows: $X_{ext}^\top =: \begin{bmatrix} X_1 & \cdots & X_{\mathbf{n}} \end{bmatrix}$, where $X_j \in \mathbb{R}^{d\times L}$. Then Γ has the block-banded Toeplitz structure (3.1) *with $s = \mathbf{n} - 1$ and with*

$$\Gamma_k = \sum_{j=1}^{\mathbf{n}-k} \mathbf{X}_j \mathbf{X}_{j+k}^\top, \quad \textit{where} \quad \mathbf{X}_k := I_K \otimes X_k. \tag{3.4}$$

Proof. Define the residual $R := \mathcal{S}(\Delta p) X_{\text{ext}}$ and the partitioning $R^\top =: \begin{bmatrix} R_1 \cdots R_{\mathbf{m}} \end{bmatrix}$, where $R_1 \in \mathbb{R}^{d\times K}$. Let $\Delta C := \mathcal{S}(\Delta p)$, with blocks ΔC_i. We have

$$\mathcal{S}(p-\Delta p)X_{\text{ext}} = 0 \iff \mathcal{S}(\Delta p)X_{\text{ext}} = \mathcal{S}(p)X_{\text{ext}}$$

$$\iff \begin{bmatrix} X_1 & X_2 & \cdots & X_{\mathbf{n}} & & & \\ & X_1 & X_2 & \cdots & X_{\mathbf{n}} & & \\ & & \ddots & \ddots & & \ddots & \\ & & & X_1 & X_2 & \cdots & X_{\mathbf{n}} \end{bmatrix} \begin{bmatrix} \Delta C_1^\top \\ \Delta C_2^\top \\ \vdots \\ \Delta C_{\mathbf{m}+\mathbf{n}-1}^\top \end{bmatrix} = \begin{bmatrix} R_1^\top \\ R_2^\top \\ \vdots \\ R_{\mathbf{m}}^\top \end{bmatrix}$$

$$\iff \underbrace{\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_{\mathbf{n}} & & & \\ & \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_{\mathbf{n}} & & \\ & & \ddots & \ddots & & \ddots & \\ & & & \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_{\mathbf{n}} \end{bmatrix}}_{G(X)} \underbrace{\begin{bmatrix} \operatorname{vec}(\Delta C_1^\top) \\ \operatorname{vec}(\Delta C_2^\top) \\ \vdots \\ \operatorname{vec}(\Delta C_{\mathbf{m}+\mathbf{n}-1}^\top) \end{bmatrix}}_{\Delta p} = \underbrace{\begin{bmatrix} \operatorname{vec}(R_1^\top) \\ \operatorname{vec}(R_2^\top) \\ \vdots \\ \operatorname{vec}(R_{\mathbf{m}}^\top) \end{bmatrix}}_{r(X)}.$$

Therefore, $\Gamma = GG^\top$ has the structure (3.1), with Γ_k's given by (3.4). ∎

The derivation of the Γ matrix for an STLS problem with block-Toeplitz data matrix is analogous to the one for an STLS problem with block-Hankel data matrix. We state the result in the next lemma.

LEMMA 3.5. *Let*

$$\mathcal{S}(p) := \begin{bmatrix} C_{\mathbf{n}} & C_{\mathbf{n}-1} & \cdots & C_1 \\ C_{\mathbf{n}+1} & C_{\mathbf{n}} & \cdots & C_2 \\ \vdots & \vdots & & \vdots \\ C_{\mathbf{m}+\mathbf{n}-1} & C_{\mathbf{m}+\mathbf{n}-2} & \cdots & C_{\mathbf{m}} \end{bmatrix} \in \mathbb{R}^{m\times(n+d)},$$

with the blocks C_i defined as in Lemma 3.4. *Then* Γ *has the block-banded Toeplitz structure* (3.1) *with* $s = \mathbf{n} - 1$ *and with*

$$\Gamma_k = \sum_{j=k+1}^{\mathbf{n}} \mathbf{X}_j \mathbf{X}_{j-k}^\top. \tag{3.5}$$

Proof. Following the same derivation as in the proof of Lemma 3.4, we find that

$$G = \begin{bmatrix} \mathbf{X_n} & \mathbf{X_{n-1}} & \cdots & \mathbf{X}_1 & & & \\ & \mathbf{X_n} & \mathbf{X_{n-1}} & \cdots & \mathbf{X}_1 & & \\ & & \ddots & \ddots & & \ddots & \\ & & & \mathbf{X_n} & \mathbf{X_{n-1}} & \cdots & \mathbf{X}_1 \end{bmatrix}.$$

Therefore, $\Gamma = GG^\top$ has the structure (3.1), with Γ_k's given by (3.5). ∎

Proof of Theorem 3.1. Lemmas 3.2–3.5 show that the weight matrix Γ for the original problem has the block-banded Toeplitz structure (3.1) with $s = \max_{l=1,\ldots,q}(\mathbf{n}_l - 1)$, where $\mathbf{n}_l$ is the number of block columns in the lth block of the data matrix. ∎

Apart from revealing the structure of Γ, the proof of Theorem 3.1 gives an algorithm for the construction of the blocks $\Gamma_0, \ldots, \Gamma_s$ that define Γ:

$$\Gamma_k = \sum_{l=1}^{q} \Gamma_k^{(l)}, \text{ where } \Gamma_k^{(l)} = \begin{cases} \sum_{j=k+1}^{\mathbf{n}_l} \mathbf{X}_j^{(l)} \mathbf{X}_{j-k}^{(l)\top} & \text{if } C^{(l)} \text{ is block-Toeplitz,} \\ \sum_{j=1}^{\mathbf{n}_l - k} \mathbf{X}_j^{(l)} \mathbf{X}_{j+k}^{(l)\top} & \text{if } C^{(l)} \text{ is block-Hankel,} \\ 0_{dK} & \text{if } C^{(l)} \text{ is exact, or} \\ \delta_k I_K \otimes (X_{\text{ext}}^{(l)\top} X_{\text{ext}}^{(l)}) & \text{if } C^{(l)} \text{ is unstructured,} \end{cases} \tag{3.6}$$

where δ is the Kronecker delta function: $\delta_0 = 1$ and $\delta_k = 0$ for $k \neq 0$.

Corollary 3.6 (positive definiteness of the weight matrix Γ). *Assume that the structure of* $\mathcal{S}$ *is given by* (2.6) *with the block* $C^{(q)}$ *being block-Toeplitz, block-Hankel, or unstructured and having at least* d *columns. Then the matrix* $\Gamma(X)$ *is positive definite for all* $X \in \mathbb{R}^{n \times d}$.

Proof. We will show that $\Gamma^{(q)}(X) > 0$ for all $X \in \mathbb{R}^{n \times d}$. From (3.2), it follows that Γ has the same property. By the assumption $\operatorname{col\,dim}(C^{(q)}) \geq d$, it follows that $X_{\text{ext}}^{(q)} = \left[\begin{smallmatrix} * \\ -I_d \end{smallmatrix}\right]$, where the $*$ denotes a block (possibly empty) depending on X. In the unstructured case, $\Gamma^{(q)} = I_m \otimes (X_{\text{ext}}^{(q)\top} X_{\text{ext}}^{(q)})$; see (3.6). But $\operatorname{rank}(X_{\text{ext}}^{(q)\top} X_{\text{ext}}^{(q)}) = d$, so that $\Gamma^{(q)}$ is nonsingular. In the block-Hankel/Toeplitz case, $G^{(q)}$ is block-Toeplitz and block-banded; see Lemmas 3.4 and 3.5. One can verify by inspection that independent of X, $G^{(q)}(X)$ has full row rank due to its row echelon form. Then $\Gamma^{(q)} = G^{(q)} G^{(q)\top} > 0$. ∎

The positive definiteness of Γ is studied in a statistical setting in [10, sect. 4], where more general conditions are given. The restriction of (2.6) that ensures $\Gamma > 0$ is fairly minor, so that in what follows we will consider STLS problems of this type and replace the pseudoinverse in (2.5) with the inverse.

In the next section, we give an interpretation of Theorem 3.1 from a statistical point of view, and in section 5 we consider in more detail the algorithmic side of the problem.

4. Stochastic interpretation. Our work on the STLS problem has its origin in the field of estimation theory. A linear multivariate *errors-in-variables* (EIV) model is defined as follows:

$$AX \approx B, \quad \text{where} \quad A = \bar{A} + \tilde{A}, \quad B = \bar{B} + \tilde{B}, \quad \text{and} \quad \bar{A}\bar{X} = \bar{B}. \tag{4.1}$$

The observations A and B are obtained from (nonstochastic) *true values* $\bar{A}$ and $\bar{B}$ with *measurement errors* $\tilde{A}$ and $\tilde{B}$ that are zero mean random matrices. Define the extended matrix $\tilde{C} := \begin{bmatrix}\tilde{A} & \tilde{B}\end{bmatrix}$ and the vector $\tilde{c} := \operatorname{vec}(\tilde{C}^\top)$ of the measurement errors. It is well known (see [25, Chap. 8]) that the TLS problem (1.1) provides a consistent estimator for the true value of the parameter $\bar{X}$ in the EIV model (4.1) if $\operatorname{cov}(\tilde{c}) = \sigma^2 I$ (and additional technical conditions are satisfied). If, in addition to $\operatorname{cov}(\tilde{c}) = \sigma^2 I$, $\tilde{c}$ is normally distributed, i.e., $\tilde{c} \sim \mathrm{N}(0, \sigma^2 I)$, then the solution $\hat{X}_{\mathrm{tls}}$ of the TLS problem is the maximum likelihood estimate of $\bar{X}$.

The EIV model (4.1) is called the *structured errors-in-variables model* if the observed data C and the true value $\bar{C} := \begin{bmatrix}\bar{A} & \bar{B}\end{bmatrix}$ have a structure defined by a function $\mathcal{S}$. Therefore,

$$C = \mathcal{S}(p) \quad \text{and} \quad \bar{C} = \mathcal{S}(\bar{p}),$$

where $\bar{p} \in \mathbb{R}^{n_p}$ is a (nonstochastic) true value of the parameter p. As a consequence the matrix of measurement errors is also structured. Let $\mathcal{S}$ be affine (2.2). Then

$$\tilde{C} = \sum_{i=1}^{n_p} S_i \tilde{p}_i \qquad \text{and} \qquad p = \bar{p} + \tilde{p},$$

where the random vector $\tilde{p}$ represents the measurement error on the structure parameter $\bar{p}$. In [10], it is proven that the STLS problem (2.1) provides a consistent estimator for the true value of the parameter $\bar{X}$ if $\operatorname{cov}(\tilde{p}) = \sigma^2 I$ (and additional technical conditions are satisfied). If $\tilde{p} \sim \mathrm{N}(0, \sigma^2 I)$, then a solution $\hat{X}$ of the STLS problem is a maximum likelihood estimate of $\bar{X}$.

Let $\tilde{r}(X) := \operatorname{vec}\big(\mathcal{S}(\tilde{p}) X_{\mathrm{ext}}\big)$ be the random part of the residual r. In the stochastic setting, the weight matrix Γ is up to the scale factor σ^2 equal to the covariance matrix $V_{\tilde{r}} := \operatorname{cov}(\tilde{r})$. Indeed, $\tilde{r} = G\tilde{p}$, so that

$$V_{\tilde{r}} := \mathbf{E}\, \tilde{r}\tilde{r}^\top = G\, \mathbf{E}\, (\tilde{p}\tilde{p}^\top) G^\top = \sigma^2 G G^\top = \sigma^2 \Gamma.$$

Next we show that the structure of Γ is in a one-to-one correspondence with the structure of $V_{\tilde{c}} := \operatorname{cov}(\tilde{c})$. Let $\Gamma_{ij} \in \mathbb{R}^{dK \times dK}$ be the (i,j)th block of Γ and let $V_{\tilde{\mathbf{c}},ij} \in \mathbb{R}^{(n+d)K \times (n+d)K}$ be the (i,j)th block of $V_{\tilde{c}}$. Define also the following partitionings of the vectors $\tilde{r}$ and $\tilde{c}$:

$$\tilde{r} =: \begin{bmatrix}\tilde{\mathbf{r}}_1 \\ \vdots \\ \tilde{\mathbf{r}}_{\mathbf{m}}\end{bmatrix}, \quad \tilde{\mathbf{r}}_i \in \mathbb{R}^{dK} \qquad \text{and} \qquad \tilde{c} =: \begin{bmatrix}\tilde{\mathbf{c}}_1 \\ \vdots \\ \tilde{\mathbf{c}}_{\mathbf{m}}\end{bmatrix}, \quad \tilde{\mathbf{c}}_i \in \mathbb{R}^{(n+d)K},$$

where $\mathbf{m} := m/K$. Using $\mathbf{r}_i = \mathbf{X}_{\mathrm{ext}} \mathbf{c}_i$, where $\mathbf{X}_{\mathrm{ext}} := (I_K \otimes X_{\mathrm{ext}}^\top)$, we have

$$\sigma^2 \Gamma_{ij} = \mathbf{E}\, \tilde{\mathbf{r}}_i \tilde{\mathbf{r}}_j^\top = \mathbf{X}_{\mathrm{ext}} \mathbf{E}\, (\tilde{\mathbf{c}}_i \tilde{\mathbf{c}}_j^\top) \mathbf{X}_{\mathrm{ext}}^\top = \mathbf{X}_{\mathrm{ext}} V_{\tilde{\mathbf{c}},ij} \mathbf{X}_{\mathrm{ext}}^\top. \tag{4.2}$$

The one-to-one relation between the structures of Γ and $V_{\tilde{c}}$ allows us to relate the structural properties of Γ, established in Theorem 3.1, with statistical properties of the measurement errors. Define stationarity and s-dependence of a centered sequence of random vectors $\tilde{\mathbf{c}} := \{\tilde{\mathbf{c}}_1, \tilde{\mathbf{c}}_2, \dots\}$, $\tilde{\mathbf{c}}_i \in \mathbb{R}^{(n+d)K}$ as follows:

- $\tilde{\mathbf{c}}$ is *stationary* if the covariance matrix $V_{\tilde{c}}$ is block-Toeplitz with block size $(n+d)K \times (n+d)K$.

- $\tilde{\mathbf{c}}$ is *s-dependent* if the covariance matrix $V_{\tilde{c}}$ is block-banded with block size $(n+d)K \times (n+d)K$ and block bandwidth $2s+1$.

The sequence of measurement errors $\tilde{\mathbf{c}}$ being stationary and s-dependent corresponds to Γ being block-Toeplitz and block-banded.

The statistical setting gives an insight into the relation between the structure of the weight matrix Γ and the structure of the data matrix C. It can be verified that the structure specification (2.6) implies stationarity and s-dependence for $\tilde{\mathbf{c}}$. This indicates an alternative (statistical) proof of Theorem 3.1; see the technical report [14].

The blocks of Γ are quadratic functions of X, $\Gamma_{ij}(X) = \mathbf{X}_{\text{ext}} W_{\tilde{\mathbf{c}},ij} \mathbf{X}_{\text{ext}}^\top$, where $W_{\tilde{\mathbf{c}},ij} := V_{\tilde{\mathbf{c}},ij}/\sigma^2$; see (4.2). Moreover, by Theorem 3.1, we have that under assumption (2.6), $W_{\tilde{\mathbf{c}},ij} = W_{\tilde{\mathbf{c}},|i-j|}$ for certain matrices $W_{\tilde{\mathbf{c}},k}$, $k = 1, \ldots, \mathbf{m}$, and $W_{\tilde{\mathbf{c}},ij} = 0$ for $|i-j| > s$, where s is defined in Theorem 3.1. Therefore,

$$\Gamma_k(X) = \mathbf{X}_{\text{ext}} W_{\tilde{\mathbf{c}},k} \mathbf{X}_{\text{ext}}^\top \quad \text{for } k = 0, 1, \ldots, s, \quad \text{where} \quad W_{\tilde{\mathbf{c}},k} := \frac{1}{\sigma^2} V_{\tilde{\mathbf{c}},k}.$$

In (3.6) we show how the matrices $\{\Gamma_k\}_{k=0}^s$ can be determined from the structure specification (2.6). Similar expressions can be written for the matrices $\{W_{\tilde{\mathbf{c}},k}\}_{k=0}^s$.

In the computational algorithm described in section 5, we use the partitioning of the matrix Γ into blocks of size $d \times d$. Let $\Gamma_{ij} \in \mathbb{R}^{d \times d}$ be the (i,j)th block of Γ and let $V_{\tilde{c},ij} \in \mathbb{R}^{(n+d)\times(n+d)}$ be the (i,j)th block of $V_{\tilde{c}}$. Define the following partitionings of the vectors $\tilde{r}$ and $\tilde{c}$:

$$\tilde{r} =: \begin{bmatrix} \tilde{r}_1 \\ \vdots \\ \tilde{r}_m \end{bmatrix}, \quad r_i \in \mathbb{R}^d \qquad \text{and} \qquad \tilde{c} =: \begin{bmatrix} \tilde{c}_1 \\ \vdots \\ \tilde{c}_m \end{bmatrix}, \quad c_i \in \mathbb{R}^{n+d}.$$

Using $r_i = X_{\text{ext}}^\top c_i$, we have

$$\Gamma_{ij} = \frac{1}{\sigma^2} \mathbf{E}\, \tilde{r}_i \tilde{r}_j^\top = \frac{1}{\sigma^2} X_{\text{ext}}^\top \mathbf{E}\, (\tilde{c}_i \tilde{c}_j^\top) X_{\text{ext}} = \frac{1}{\sigma^2} X_{\text{ext}}^\top V_{\tilde{c},ij} X_{\text{ext}} =: X_{\text{ext}}^\top W_{\tilde{c},ij} X_{\text{ext}}.$$

5. Efficient cost function and first derivative evaluation. We consider an efficient numerical method for solving the STLS problem (2.1) by applying standard local optimization algorithms to the equivalent problem (2.5). With this approach, the main computational effort is in the cost function and its first derivative evaluation.

First, we describe the evaluation of the cost function: given X, compute $f_0(X)$. For given X, and with $\{\Gamma_k\}_{k=0}^s$ constructed as described in the proof of Theorem 3.1, the weight matrix $\Gamma(X)$ is specified. Then from the solution of the system $\Gamma(X) y_r(X) = r(X)$, the cost function is found as $f_0(X) = r^\top(X) y_r(X)$.

The properties of $\Gamma(X)$ can be exploited in the solution of the system $\Gamma y_r = r$. The subroutine `MB02GD` from the SLICOT library [2] exploits both the block-Toeplitz and the banded structure to compute a Cholesky factor of Γ in $O((dK)^2 sm)$ flops. In combination with the LAPACK subroutine `DPBTRS` that solves block-banded triangular systems of equations, the cost function is evaluated in $O(m)$ flops. Thus an algorithm for local optimization that uses only cost function evaluations has computational complexity $O(m)$ flops per iteration, because the computations needed internally for the optimization algorithm do not depend on m.

Next, we describe the evaluation of the derivative. The derivative of the cost function f_0 is (see the appendix)

$$f_0'(X) = 2 \sum_{i,j=1}^{m} a_j r_i^\top(X) M_{ij}(X) - 2 \sum_{i,j=1}^{m} \begin{bmatrix} I & 0 \end{bmatrix} W_{\tilde{c},ij} \begin{bmatrix} X \\ -I \end{bmatrix} N_{ji}(X), \tag{5.1}$$

where $A^\top =: \begin{bmatrix} a_1 & \cdots & a_m \end{bmatrix}$, with $a_i \in \mathbb{R}^n$,

$$M(X) := \Gamma^{-1}(X), \qquad N(X) := \Gamma^{-1}(X) r(X) r^\top(X) \Gamma^{-1}(X),$$

and $M_{ij} \in \mathbb{R}^{d \times d}$, $N_{ij} \in \mathbb{R}^{d \times d}$ are the (i,j)th blocks of M and N, respectively.

Consider the two partitionings of $y_r \in \mathbb{R}^{md}$,

$$y_r =: \begin{bmatrix} y_{r,1} \\ \vdots \\ y_{r,m} \end{bmatrix}, \quad y_{r,i} \in \mathbb{R}^d \qquad \text{and} \qquad y_r =: \begin{bmatrix} \mathbf{y}_{r,1} \\ \vdots \\ \mathbf{y}_{r,\mathbf{m}} \end{bmatrix}, \quad \mathbf{y}_{r,i} \in \mathbb{R}^{dK}, \tag{5.2}$$

where $\mathbf{m} := m/K$. The first sum in (5.1) becomes

$$\sum_{i,j=1}^{m} a_j r_i^\top M_{ij} = A^\top Y_r, \quad \text{where} \quad Y_r^\top := \begin{bmatrix} y_{r,1} & \cdots & y_{r,m} \end{bmatrix}. \tag{5.3}$$

Define the sequence of matrices

$$\mathbf{N}_k := \sum_{i=1}^{\mathbf{m}-k} \mathbf{y}_{r,i+k} \mathbf{y}_{r,i}^\top, \quad \mathbf{N}_k = \mathbf{N}_{-k}^\top, \quad k = 0, \ldots, s.$$

The second sum in (5.1) can be written as

$$\sum_{i,j=1}^{m} \begin{bmatrix} I & 0 \end{bmatrix} W_{\tilde{c},ij} \begin{bmatrix} X \\ -I \end{bmatrix} N_{ji} = \sum_{k=-s}^{s} \sum_{i,j=1}^{K} (W_{\tilde{\mathbf{a}},k,ij} X - W_{\tilde{\mathbf{a}}\tilde{\mathbf{b}},k,ij}) \mathbf{N}_{k,ij}^\top,$$

where $W_{\tilde{\mathbf{c}},k,ij} \in \mathbb{R}^{(n+d)\times(n+d)}$ is the (i,j)th block of $W_{\tilde{\mathbf{c}},k} \in \mathbb{R}^{K(n+d)\times K(n+d)}$, $W_{\tilde{\mathbf{a}},k,ij} \in \mathbb{R}^{n \times n}$ and $W_{\tilde{\mathbf{a}}\tilde{\mathbf{b}},k,ij} \in \mathbb{R}^{n \times d}$ are defined as blocks of $W_{\tilde{\mathbf{c}},k,ij}$ as

$$W_{\tilde{\mathbf{c}},k,ij} =: \begin{bmatrix} W_{\tilde{\mathbf{a}},k,ij} & W_{\tilde{\mathbf{a}}\tilde{\mathbf{b}},k,ij} \\ W_{\tilde{\mathbf{b}}\tilde{\mathbf{a}},k,ij} & W_{\tilde{\mathbf{b}},k,ij} \end{bmatrix},$$

and $\mathbf{N}_{k,ij} \in \mathbb{R}^{d \times d}$ is the (i,j)th block of $\mathbf{N}_k \in \mathbb{R}^{dK \times dK}$.

Thus the evaluation of the derivative $f_0'(X)$ uses the solution of $\Gamma y_r = r$, already computed for the cost function evaluation and additional operations of $O(m)$ flops. The steps described above are summarized in Algorithm 1.

Algorithm 1. Cost function and first derivative evaluation.

1: Input: A, B, X, $\{W_{\tilde{\mathbf{c}},k}\}_{k=0}^{s}$.
2: $\Gamma_k = (I_K \otimes X_{\text{ext}}^\top) W_{\tilde{\mathbf{c}},k} (I_K \otimes X_{\text{ext}}^\top)^\top$ for $k = 0, 1, \ldots, s$,
3: $r = \operatorname{vec}\big((AX - B)^\top\big)$,
4: solve (via `MB02GD` and `DPBTRS`) $\Gamma y_r = r$, where Γ is given in (3.1),
5: $f_0 = r^\top y_r$.
6: If only the cost function evaluation is required, output f_0 and stop.
7: $\mathbf{N}_k = \sum_{i=1}^{\mathbf{m}-k} \mathbf{y}_{r,i+k} \mathbf{y}_{r,i}^\top$ for $k = 0, 1, \ldots, s$, where $\mathbf{y}_i$ is defined in (5.2).
8: $f_0' = 2A^\top Y_r - 2 \sum_{k=-s}^{s} \sum_{i,j=1}^{K} (W_{\tilde{\mathbf{a}},k,ij} X - W_{\tilde{\mathbf{a}}\tilde{\mathbf{b}},k,ij}) \mathbf{N}_{k,ij}^\top$, where Y_r is defined in (5.3).
9: Output f_0, f_0' and stop.

The flops per step for Algorithm 1 are as follows:

2. $(n+d)(n+2d)dK^3$.
3. $m(n+1)d$.
4. msd^2K^2.
5. md.
7. $msd^2K - s(s+1)d^2K^2/2$.
8. $mnd + (2s+1)(nd+n+1)dK^2$.

Thus in total $O(md(sdK^2+n) + n^2dK^3 + 3nd^2K^3 + 2d^3K^3 + 2snd^2K^2)$ flops are required for cost function and first derivative evaluation. Note that the flop counts depend on the structure through s.

Using the computation of the cost function and its first derivative, as outlined above, we can apply the BFGS (Broyden, Fletcher, Goldfarb, and Shanno) quasi-Newton method. A more efficient alternative, however, is to apply a nonlinear least squares optimization algorithm, such as the Levenberg–Marquardt algorithm. Let $\Gamma = U^\top U$ be the Cholesky factorization of Γ. Then $f = F^\top F$, with $F := U^{-1}r$. (Note that the evaluation of $F(X)$ is cheaper than that of $f(X)$.) We do not know an analytic expression for the Jacobian matrix $J(X) = [\partial F_i/\partial x_j]$, but instead we use the so-called pseudo-Jacobian J_+ proposed in [9]. The evaluation of J_+ can be done efficiently, using the approach described above for $f'(X)$.

Moreover, by using the nonlinear least squares approach and the pseudo-Jacobian J_+, we have as a byproduct of the optimization algorithm an estimate of the covariance matrix $V_{\hat{x}} = \mathbf{E}\left(\mathrm{vec}(\hat{X})\mathrm{vec}^\top(\hat{X})\right)$. As shown in [19, Chap. 17.4.7, eqns. (17)–(35)], $V_{\hat{x}} \approx \left(J_+^\top(\hat{X})J_+(\hat{X})\right)^{-1}$. Using $V_{\hat{x}}$, we can compute statistical confidence bounds for the estimate $\hat{X}$.

6. Applications and simulation examples. Under assumption (2.6), the specification of $\mathcal{S}$ is given by K and the array $\mathcal{D} \in \{(\mathtt{T},\mathtt{H},\mathtt{U},\mathtt{E}) \times \mathbb{N} \times \mathbb{N}\}^q$ that describes the structure of the blocks $\{C^{(l)}\}_{l=1}^q$; $\mathcal{D}_l$ specifies the block $C^{(l)}$ by giving its type $\mathcal{D}_l(1)$ ($\mathtt{T}$ = block-Toeplitz, $\mathtt{H}$ = block-Hankel, $\mathtt{U}$ = unstructured, and $\mathtt{E}$ = exact), the number of columns $n_l = \mathcal{D}_l(2)$, and, for block-Toeplitz/Hankel blocks, the column dimension $\mathcal{D}_l(3)$ of the block. The following well-known problems are special cases of the block-Toeplitz/Hankel STLS problem of this paper for particular choices of the structure description $\mathcal{D}$. (If not specified, K and the third element of $\mathcal{D}_l$ are equal to one.)

1. *Least squares problem:* $AX \approx B$, $A \in \mathbb{R}^{m\times n}$ exact, $B \in \mathbb{R}^{m\times d}$ noisy and unstructured is achieved by $\mathcal{D} = [[\mathtt{E}\ n], [\mathtt{U}\ d]]$.

2. *TLS problem:* $AX \approx B$, $C = [A\ B] \in \mathbb{R}^{m\times(n+d)}$ noisy and unstructured is achieved by $\mathcal{D} = [\mathtt{U}\ n+d]$.

3. *Data least squares problem* [3]: $AX \approx B$, $A \in \mathbb{R}^{m\times n}$ noisy and unstructured, and $B \in \mathbb{R}^{m\times d}$ exact is achieved by $\mathcal{D} = [[\mathtt{U}\ n], [\mathtt{E}\ d]]$.

4. *Mixed LS-TLS problem* [25, sect. 3.5]: $AX \approx B$, $A = [A_{\mathrm{noisy}}\ A_{\mathrm{exact}}]$, $A_{\mathrm{noisy}} \in \mathbb{R}^{m\times n_1}$ and $B \in \mathbb{R}^{m\times d}$ noisy and unstructured, $A_{\mathrm{exact}} \in \mathbb{R}^{m\times n_2}$ exact is achieved by $\mathcal{D} = [[\mathtt{U}\ n_1], [\mathtt{E}\ n_2], [\mathtt{U}\ d]]$.

5. *Hankel low-rank approximation problem* [4, sect. 4.5], [21]:

$$\min_{\Delta p} \|\Delta p\|^2 \quad \text{subject to} \quad \mathcal{H}(p - \Delta p) \text{ has given rank } n, \tag{6.1}$$

where $\mathcal{H}$ is a mapping from the parameter space $\mathbb{R}^{n_p}$ to the set of the $m \times (n+d)$ block-Hankel matrices, with block size $n_y \times n_u$. If the rank constraint is expressed as

$\mathcal{H}(\hat{p}) \left[\begin{smallmatrix} X \\ -I \end{smallmatrix} \right] = 0$, with $X \in \mathbb{R}^{n \times d}$ an additional variable, then (6.1) becomes an STLS problem with $K = n_y$ and $\mathcal{D} = \{[\mathtt{H}\ n + d\ n_u]\}$.

6. *Deconvolution problem:* For a description of the problem and its formulation as an STLS problem, see [17]. In [17] a finite impulse response (FIR) filter identification problem is considered, which is an application of deconvolution for system identification. The structure in this case is $\mathcal{D} = [[\mathtt{T}\ n], [\mathtt{U}\ 1]]$, where n is the number of lags of the FIR filter.

7. *Transfer function estimation:* For a description of the problem and its formulation as an STLS problem, see [4, sect. 4.6]. The structure arising in this problem is $\mathcal{D} = [[\mathtt{H}\ n_b + 1], [\mathtt{H}\ n_a + 1]]$, where n_b is the order of the numerator and n_a is the order of the denominator of the estimated transfer function.

The last three problems have system theoretic interpretation—the Hankel–low-rank approximation problem is a *noisy realization problem* [5] or alternatively a model reduction problem (see section 6.2), and the deconvolution and the transfer function estimation problems are system identification problems (see section 6.1). For multi-input, multi-output (MIMO) systems, these problems result in block-Toeplitz/Hankel structured matrices.

Next we show simulation examples for the system identification and model reduction applications. They aim to illustrate the applicability of the derived algorithm for real-life problems. More details on the application of STLS for these problems and more realistic identification examples can be found in [16].

6.1. Improvement of the subspace identification estimate. Maximum likelihood SISO transfer function identification from noisy input/output data can be formulated as an STLS problem with a data matrix composed of two Hankel or Toeplitz structured blocks next to each other; see [4, sect. 4.6]. The STLS method, however, needs a good initial approximation. On the other hand, the popular subspace identification methods [26] do not need initial approximation but do not minimize a particular cost function. As a result, in general, they are statistically not as accurate as the methods based on the maximum likelihood principle. A natural idea is to use the subspace method estimate, on a second stage of the estimation problem, as an initial approximation for the STLS method. The latter is expected to reduce the estimation error.

We show a simulation example to illustrate the idea. Consider the linear time-invariant (LTI) system with a transfer function

$$\bar{H}(z) = 0.151 \cdot \frac{1 + 0.9z + 0.49z^2 + 0.145z^3}{1 - 1.2z + 0.81z^2 - 0.27z^3}.$$

This is the "true model" that we aim to identify. Let $(\bar{u}(t), \bar{y}(t))_{t=1}^{m}$ be an input/output trajectory of the system, where $\bar{u}$ is a zero mean, white process with unit variance. The data available for the identification are $(u(t), y(t))_{t=1}^{m}$, where $u = \bar{u} + \tilde{u}$, $y = \bar{y} + \tilde{y}$, and $\tilde{u}$, $\tilde{y}$ are zero mean, normal, white, measurement noise, with variance $\sigma^2 = 0.05^2$. Assuming that the exact system order is known, we apply the state space algorithm N4SID [26]. The obtained estimate is used as an initial approximation for the STLS algorithm.

Let vec_par be an operator that stacks the parameters of a transfer function, i.e., the coefficients of the numerator and denominator, in a vector. We define the average relative error of estimation by

$$\bar{e}_{\text{par}} = \frac{1}{N} \sum_{k=1}^{N} \frac{\|\text{vec\_par}(\bar{H}) - \text{vec\_par}(\hat{H}^{(k)})\|_2}{\|\text{vec\_par}(\bar{H})\|_2}.$$

Here $\hat{H}^{(k)}$ denotes the identified transfer function in the kth repetition of the experiment; $N = 100$ repetitions of the experiment with different measurement noise realizations are performed. Figure 6.1 shows the average relative errors $\bar{e}_{\text{par}}$ for the subspace method and for the STLS-based maximum likelihood method as a function of the time horizon m. The example shows that, for large sample sizes, the two approaches give close estimates and, for small sample sizes, the subspace estimate can be improved by the STLS method.

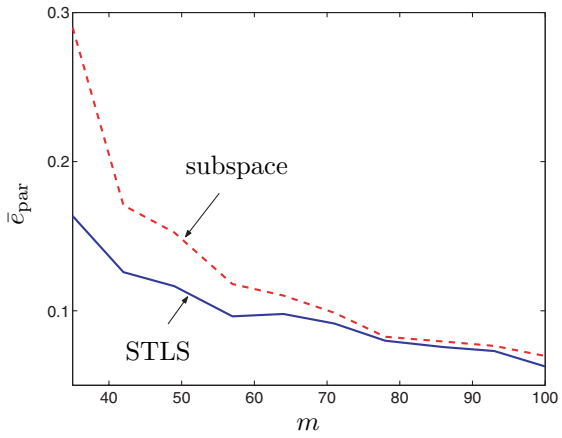


FIG. 6.1. *Results for the system identification example: Average relative error of estimation for the subspace and STLS methods.*

6.2. MIMO system model reduction. Finite horizon 2-norm optimal model reduction can be formulated as an STLS problem with a block-Hankel structured data matrix. On the other hand, balanced model reduction [18], like subspace identification, does not require initial approximation but also does not minimize a particular cost function. Again an improvement can be expected over the balanced model reduction method when the STLS method is used on a second stage of the approximation.

To illustrate the idea, consider the following example. A 10th order, 2-input, 1-output random system has to be approximated by an rth order system, where $r = 2, 4, 6, 8$. First we apply balanced reduction. The obtained solution is used as an initial approximation for the STLS method. Table 6.1 shows the average relative $\mathcal{H}_2$-errors of approximation over $N = 100$ repetitions:

$$\bar{e}_{\mathcal{H}_2} = \frac{1}{N} \sum_{k=1}^{N} \frac{\|\bar{H} - \hat{H}^{(k)}\|_{\mathcal{H}_2}}{\|\bar{H}\|_{\mathcal{H}_2}}.$$

The example confirms that the STLS method can be used to improve the result of the balanced model reduction method.

TABLE 6.1

Results for the model reduction example: Average relative error of estimation $\bar{e}_{\mathcal{H}_2}$ for balanced model reduction (BMR) and STLS.

Method	$r = 2$	$r = 4$	$r = 6$	$r = 8$
BMR	0.1062122	0.0288455	0.0012585	0.0000259
STLS	0.1034344	0.0276010	0.0012433	0.0000229

7. Conclusions. We considered an STLS problem with the structure of the data matrix, specified blockwise. Each of the blocks can be block-Toeplitz/Hankel structured, unstructured, or exact. It was shown that such a formulation is flexible and covers as special cases many previously studied structured and unstructured matrix approximation problems.

The numerical solution method of [13] was extended to the block-Toeplitz/Hankel case. The approach is based on an equivalent unconstrained optimization problem: $\min_X r^\top(X)\Gamma^{-1}(X)r(X)$. We proved that under assumption (2.6) about the structure of the data matrix, the weight matrix Γ is block-Toeplitz and block-banded. These properties were used for cost function and first derivative evaluation with computational cost linear in the sample size.

The extension to block-Toeplitz/Hankel structured matrices is motivated by identification and model reduction problems for MIMO dynamical systems. Useful further extensions are (i) to consider a weighted quadratic cost function $\Delta p^\top V \Delta p$, with $V > 0$ diagonal, and (ii) regularized STLS problems, where the cost function is augmented with the regularization term $\operatorname{vec}^\top(X)Q\operatorname{vec}(X)$. These extensions are still computable in $O(m)$ flops per iteration.

Appendix. Derivation of the first derivative of the cost function f_0. Denote by D the differential operator. It acts on a differentiable function $f_0 : U \to \mathbb{R}$, where U is an open set in $\mathbb{R}^{n\times d}$ and gives as a result another function, the differential of f_0, $D(f_0) : U \times \mathbb{R}^{n\times d} \to \mathbb{R}$. The differential $D(f_0)$ is linear in its second argument, i.e.,

$$D(f_0) := \mathrm{d}f_0(X, H) = \operatorname{trace}\big(f_0'(X)H^\top\big), \tag{A.1}$$

and has the property

$$f_0(X + H) = f_0(X) + \mathrm{d}f_0(X, H) + \mathrm{o}(\|H\|_{\mathrm{F}})$$

for all $X \in U$ and for all $H \in \mathbb{R}^{n\times d}$. (The notation $\mathrm{o}(\|H\|_{\mathrm{F}})$ has the usual meaning: $g(H) = \mathrm{o}(\|H\|_{\mathrm{F}})$ if and only if $\lim_{\|H\|_{\mathrm{F}}\to 0} g(H)/\|H\|_{\mathrm{F}} = 0$.) The function $f_0' : U \to \mathbb{R}^{n\times l}$ is the derivative of f_0. We compute it by deriving the differential $D(f_0)$ and representing it in the form (A.1), from which $f_0'(X)$ is extracted.

The differential of the cost function $f_0(X) = r^\top(X)\Gamma^{-1}(X)r(X)$ is (using the rule for differentiation of an inverse matrix)

$$\mathrm{d}f_0(X, H) = 2r^\top\Gamma^{-1}\begin{bmatrix} H^\top a_1 \\ \vdots \\ H^\top a_m \end{bmatrix} - r^\top\Gamma^{-1}\big(\mathrm{d}\Gamma(X, H)\big)\Gamma^{-1}r.$$

The differential of the weight matrix

$$\Gamma = V_{\tilde r} = \mathbf{E}\,\tilde r\tilde r^\top = \mathbf{E}\begin{bmatrix} X^\top\tilde a_1 - \tilde b_1 \\ \vdots \\ X^\top\tilde a_m - \tilde b_m \end{bmatrix}\begin{bmatrix}\tilde a_1^\top X - \tilde b_1^\top & \cdots & \tilde a_m^\top X - \tilde b_m^\top\end{bmatrix},$$

where $\tilde A^\top =: \begin{bmatrix}\tilde a_1 & \cdots & a_m\end{bmatrix}$, $\tilde a_i \in \mathbb{R}^n$, and $\tilde B^\top =: \begin{bmatrix}\tilde b_1 & \cdots & b_m\end{bmatrix}$, $\tilde b_i \in \mathbb{R}^d$, is

$$\mathrm{d}\Gamma(X, H) = \mathbf{E}\begin{bmatrix} H^\top\tilde a_1 \\ \vdots \\ H^\top\tilde a_m \end{bmatrix}\tilde r^\top + \mathbf{E}\,\tilde r\begin{bmatrix}\tilde a_1^\top H & \cdots & \tilde a_m^\top H\end{bmatrix}. \tag{A.2}$$

With $M_{ij} \in \mathbb{R}^{d\times d}$ denoting the (i,j)th block of Γ^{-1},

$$\begin{aligned}\frac{1}{2}\mathrm{d}f_0(X,H) &= \sum_{i,j=1}^{m} r_i^\top M_{ij} H^\top a_j - \sum_{i,j,k,l=1}^{m} r_l^\top M_{li} H^\top \mathbf{E}\,\tilde{a}_i \tilde{c}_j^\top X_{\mathrm{ext}} M_{jl} r_l \\ &= \mathrm{trace}\Big(\Big(\sum_{i,j=1}^{m} a_j r_i^\top M_{ij} - \sum_{i,j,k,l=1}^{m} \begin{bmatrix} I & 0 \end{bmatrix} V_{\tilde{c},ij} X_{\mathrm{ext}} M_{jl} r_l r_l^\top M_{li}\Big) H^\top\Big),\end{aligned}$$

so that

$$\frac{1}{2} f_0'(X) = \sum_{i,j=1}^{m} a_j r_i^\top M_{ij} - \sum_{i,j=1}^{m} \begin{bmatrix} I & 0 \end{bmatrix} V_{\tilde{c},ij} X_{\mathrm{ext}} N_{ji},$$

where $N_{ji}(X) := \sum_{l=1}^{m} M_{jl} r_l \cdot \sum_{l=1}^{m} r_l^\top M_{li}$.

Acknowledgments. We would like to thank A. Kukush, M. Schuermans, P. Lemmerling, N. Mastronardi, and D. Sima for helpful discussion on the STLS problem.

REFERENCES

[1] T. Abatzoglou, J. Mendel, and G. Harada, *The constrained total least squares technique and its application to harmonic superresolution*, IEEE Trans. Signal Process., 39 (1991), pp. 1070–1087.

[2] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga, *SLICOT—a subroutine library in systems and control theory*, in Applied and Computational Control, Signal and Circuits, Vol. 1, B. N. Datta, ed., Birkhäuser, Boston, 1999, Chap. 10, pp. 499–539.

[3] G. Cirrincione, G. Ganesan, K. Hari, and S. Van Huffel, *Direct and neural techniques for the data least squares problem*, in Proceedings of the International Symposium on the Mathematical Theory of Networks and Systems (MTNS), Perpignan, France, 2000.

[4] B. De Moor, *Structured total least squares and L_2 approximation problems*, Linear Algebra Appl., 188–189 (1993), pp. 163–207.

[5] B. De Moor, *Total least squares for affinely structured matrices and the noisy realization problem*, IEEE Trans. Signal Process., 42 (1994), pp. 3104–3113.

[6] B. De Moor and B. Roorda, *L_2-optimal linear system identification structured total least squares for SISO systems*, in Proceedings of the 33th Conference on Decision and Control (CDC), Lake Buena Vista, FL, IEEE Control Systems Society, 1994, pp. 2874–2879.

[7] W. A. Fuller, *Measurement Error Models*, John Wiley, New York, 1987.

[8] 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.

[9] P. Guillaume and R. Pintelon, *A Gauss–Newton-like optimization algorithm for "weighted" nonlinear least-squares problems*, IEEE Trans. Signal Process., 44 (1996), pp. 2222–2228.

[10] A. Kukush, I. Markovsky, and S. Van Huffel, *Consistency of the structured total least squares estimator in a multivariate errors-in-variables model*, J. Statist. Plann. Inference, to appear.

[11] P. Lemmerling, N. Mastronardi, and S. V. Huffel, *Fast algorithm for solving the Hankel/Toeplitz structured total least squares problem*, Numer. Algorithms, 23 (2000), pp. 371–392.

[12] M. J. Levin, *Estimation of a system pulse transfer function in the presence of noise*, IEEE Trans. Automat. Control, 9 (1964), pp. 229–235.

[13] I. Markovsky, S. Van Huffel, and A. Kukush, *On the computation of the multivariate structured total least squares estimator*, Numer. Linear Algebra Appl., 11 (2004), pp. 591–608.

[14] I. Markovsky, S. Van Huffel, and R. Pintelon, *Block-Toeplitz/Hankel Structured Total Least Squares*, Tech. Rep. 03–135, Department of Electrical Engineering, K.U. Leuven, Belgium, 2003.

[15] I. Markovsky, S. Van Huffel, and R. Pintelon, *Software for Structured Total Least Squares Estimation: User's Guide*, Tech. Rep. 03–136, Department of Electrical Engineering, K.U. Leuven, Belgium, 2003.

[16] I. Markovsky, J. C. Willems, S. Van Huffel, B. D. Moor, and R. Pintelon, *Application of Structured Total Least Squares for System Identification and Model Reduction*, Tech. Rep. 04–51, Department of Electrical Engineering, K.U. Leuven, Belgium, 2004.

[17] N. Mastronardi, P. Lemmerling, and S. Van Huffel, *Fast structured total least squares algorithm for solving the basic deconvolution problem*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 533–553.

[18] B. C. Moore, *Principal component analysis in linear systems: Controllability, observability and model reduction*, IEEE Trans. Automat. Control, 26 (1981), pp. 17–31.

[19] R. Pintelon and J. Schoukens, *System Identification: A Frequency Domain Approach*, IEEE Press, Piscataway, NJ, 2001.

[20] J. B. Rosen, H. Park, and J. Glick, *Total least norm formulation and solution for structured problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 110–126.

[21] M. Schuermans, P. Lemmerling, and S. Van Huffel, *Structured weighted low rank approximation*, Numer. Linear Algebra Appl., 11 (2004), pp. 609–618.

[22] S. Van Huffel, ed., *Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modeling*, SIAM, Philadelphia, 1997.

[23] S. Van Huffel and P. Lemmerling, eds., *Total Least Squares and Errors-in-Variables Modeling: Analysis, Algorithms and Applications*, Kluwer Academic, Dordrecht, The Netherlands, 2002.

[24] S. Van Huffel and J. Vandewalle, *Analysis and properties of the generalized total least squares problem $AX \approx B$ when some or all columns in A are subject to error*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 294–315.

[25] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*, SIAM, Philadelphia, 1991.

[26] P. Van Overschee and B. De Moor, *Subspace Identification for Linear Systems: Theory, Implementation, Applications*, Kluwer Academic, Dordrecht, The Netherlands, 1996.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 4, pp. 1100–1114

THE INVERSE EIGENPROBLEM OF CENTROSYMMETRIC MATRICES WITH A SUBMATRIX CONSTRAINT AND ITS APPROXIMATION*

ZHENG-JIAN BAI†

Abstract. In this paper, we first consider the existence of and the general expression for the solution to the constrained inverse eigenproblem defined as follows: given a set of complex n-vectors $\{\mathbf{x}_i\}_{i=1}^m$ and a set of complex numbers $\{\lambda_i\}_{i=1}^m$, and an s-by-s real matrix C_0, find an n-by-n real centrosymmetric matrix C such that the s-by-s leading principal submatrix of C is C_0, and $\{\mathbf{x}_i\}_{i=1}^m$ and $\{\lambda_i\}_{i=1}^m$ are the eigenvectors and eigenvalues of C, respectively. We are then concerned with the best approximation problem for the constrained inverse problem whose solution set is nonempty. That is, given an arbitrary real n-by-n matrix $\tilde{C}$, find a matrix C which is the solution to the constrained inverse problem such that the distance between C and $\tilde{C}$ is minimized in the Frobenius norm. We give an explicit solution and a numerical algorithm to the best approximation problem. Some illustrative experiments are also presented.

Key words. inverse problem, centrosymmetric matrix, best approximation

AMS subject classifications. 65F18, 65F15, 65F35

DOI. 10.1137/S0895479803434185

1. Introduction.

Let E_n be the n-by-n backward identity matrix, i.e., E_n has 1 on the antidiagonal and zeros elsewhere. An n-by-n real matrix C is said to be *centrosymmetric* if $C = E_nCE_n$. The centrosymmetric matrices have practical applications in many areas such as pattern recognition [10], the numerical solution of certain differential equations [1, 4], Markov processes [22], and various physical and engineering problems [11, 12]. The symmetric Toeplitz matrices, an important subclass of the class of symmetric centrosymmetric matrices, appear naturally in digital signal processing applications and other areas [13].

The inverse eigenproblems play an important role in many applications such as control theory [23], the design of Hopfield neural networks [8, 16], vibration theory [20], and structure mechanics and molecular spectroscopy [14]. For recent progress, see, for instance, [7, 25]. The inverse eigenproblem for centrosymmetric matrices has been discussed by Bai and R. Chan [2]. However, the inverse eigenproblem for centrosymmetric matrices with a submatrix constraint has not been discussed. In this paper, we will consider two related problems. The first problem is the constrained inverse eigenproblem for centrosymmetric matrices.

PROBLEM I. *Given the eigenpairs* $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m] \in \mathbb{C}^{n\times m}$, $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_m) \in \mathbb{C}^{m\times m}$, *and a matrix* $C_0 \in \mathbb{R}^{s\times s}$, *find a centrosymmetric matrix* C *in* $\mathbb{R}^{n\times n}$ *such that* $CX = X\Lambda$ *and the* s*-by-*s *leading principal submatrix of* C *is* C_0.

The prototype of this problem initially arose in the design of Hopfield neural networks [8, 16]. It also occurs in the design of vibration in mechanism, civil engineering, and aviation [5]. The problem has been studied for bisymmetric matrices in [18]. The

*Received by the editors September 2, 2003; accepted for publication (in revised form) by P. Van Dooren June 18, 2004; published electronically May 6, 2005. This research was partially supported by the Hong Kong Research Grant Council grants CUHK4243/01P and CUHK DAG 2060220.
http://www.siam.org/journals/simax/26-4/43418.html

†Department of Mathematics, Chinese University of Hong Kong, Shatin, NT, Hong Kong, China (zjbai@math.cuhk.edu.hk). Current address: Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543 (matbzj@nus.edu.sg).

second problem we consider in this paper is the problem of best approximation.

PROBLEM II. *Let $\mathcal{L}_S$ be the solution set of Problem* I. *Given a matrix $\tilde{C} \in \mathbb{R}^{n\times n}$, find $C^* \in \mathcal{L}_S$ such that*

$$\|C^* - \tilde{C}\| = \min_{C\in\mathcal{L}_S} \|C - \tilde{C}\|,$$

where $\|\cdot\|$ is the Frobenius norm.

The best approximation problem arises frequently in experimental design; see, for instance, [17, p. 123]. Here the matrix $\tilde{C}$ may be a matrix obtained from experiments, but it may not satisfy the structural requirement (centrosymmetric or the submatrix constraint) and/or spectral requirement (having eigenpairs X and Λ). The best estimate C^* is the matrix that satisfies both requirements and is the best approximation of $\tilde{C}$ in the Frobenius norm. In addition, because there are fast algorithms for solving various kinds of centrosymmetric matrices [15], the best approximate C^* of $\tilde{C}$ can also be used as a preconditioner in the preconditioned conjugate gradient method for solving linear systems with coefficient matrix $\tilde{C}$; see, for instance, [3].

We remark that when $s = 0$, Problem I is reduced to the inverse eigenproblem for centrosymmetric matrices discussed by Bai and R. Chan [2]. When $s = n$, $C^* = C_0$ is the best approximation of the matrix $\tilde{C}$ to Problem II. In this paper, we consider the general case when $0 < s < n$.

In this paper, we use the following notations. We denote the identity matrix of order n by I_n. Let $\mathrm{rank}(A)$ be the rank of a matrix A. Let A^+ and $A(1:s)$ denote the Moore–Penrose generalized inverse and the leading principal submatrix of a matrix A, respectively. $\mathcal{R}(A)$ and $\mathcal{N}(A)$ denote the column space and the null space of A, respectively.

The paper is organized as follows. In section 2 we first review the structure of centrosymmetric matrices and give some useful lemmas. In section 3 we provide the solvability conditions for and the general solutions of Problem I. In section 4 we show the existence and uniqueness of the solution to Problem II when the solution set of Problem I is nonempty, derive a formula for the best approximation of Problem II, and then propose a numerical algorithm for computing the minimizer. In section 5 an experiment is presented to illustrate our results. Finally, in section 6, we give some conclusions.

2. Preliminary lemmas. In this section, we will recall the properties of centrosymmetric matrices and give some preliminary lemmas.

Let $k = [n/2]$ denote the largest integer number that is not greater than $n/2$. When $n = 2k$, we define

$$K = \frac{1}{\sqrt{2}} \begin{pmatrix} I_k & I_k \\ E_k & -E_k \end{pmatrix},$$

and when $n = 2k + 1$, let

$$K = \frac{1}{\sqrt{2}} \begin{pmatrix} I_k & \mathbf{0} & I_k \\ \mathbf{0} & \sqrt{2} & \mathbf{0} \\ E_k & \mathbf{0} & -E_k \end{pmatrix}.$$

Clearly, K is orthogonal. Then we have the following splitting of centrosymmetric matrices into smaller submatrices using K; see, for example, [9, 2].

LEMMA 1 (see [9]). *Denote the set of all n-by-n real centrosymmetric matrices by $\mathcal{C}_n$. Then any $C \in \mathcal{C}_{2k}$ can be written as*

$$C = \begin{pmatrix} A & BE_k \\ E_kB & E_kAE_k \end{pmatrix} = K \begin{pmatrix} A+B & 0 \\ 0 & A-B \end{pmatrix} K^T, \quad A, B \in \mathbb{R}^{k\times k}.$$

Any $C \in \mathcal{C}_{2k+1}$ can be written as

$$C = \begin{pmatrix} A & \mathbf{p} & BE_k \\ \mathbf{q}^T & c & \mathbf{q}^TE_k \\ E_kB & E_k\mathbf{p} & E_kAE_k \end{pmatrix} = K \begin{pmatrix} A+B & \sqrt{2}\mathbf{p} & 0 \\ \sqrt{2}\mathbf{q}^T & c & 0 \\ 0 & 0 & A-B \end{pmatrix} K^T,$$

where $A, B \in \mathbb{R}^{k\times k}$, $\mathbf{p}, \mathbf{q} \in \mathbb{R}^k$, $c \in \mathbb{R}$. Moreover, for all $n = 2k$ and $2k+1$, any $C \in \mathcal{C}_n$ is of the form

$$C = K \begin{pmatrix} F_1 & 0 \\ 0 & F_2 \end{pmatrix} K^T, \quad F_1 \in \mathbb{R}^{(n-k)\times(n-k)}, F_2 \in \mathbb{R}^{k\times k}. \tag{1}$$

LEMMA 2. *Suppose that $C \in \mathcal{C}_n$ and $C_0 = C(1:s)$. If $s < n-k$, then*

$$F_1(1:s) + F_2(1:s) = 2C_0, \tag{2}$$

where F_1 and F_2 are the same as (1), and if $s \geq n-k$, then we obtain

$$C = \begin{pmatrix} C_{11} & C_{12} & HE_{n-s} \\ C_{21} & C_{22} & E_{2s-n}C_{21}E_{n-s} \\ E_{n-s}H & E_{n-s}C_{12}E_{2s-n} & E_{n-s}A_{11}E_{n-s} \end{pmatrix}, \tag{3}$$

where $H \in \mathbb{R}^{(n-s)\times(n-s)}$ and $C_0 = C(1:s) = \left(\begin{smallmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{smallmatrix}\right)$ with $C_{11} \in \mathbb{R}^{(n-s)\times(n-s)}$ and $C_{22} \in \mathcal{C}_{2s-n}$.

Proof. If $s < n-k$, we get from Lemma 1 that

$$C(1:s) = A(1:s)$$

and

$$F_1(1:s) + F_2(1:s) = (A+B)(1:s) + (A-B)(1:s) = 2A(1:s).$$

Thus (2) holds.

If $s \geq n-k$, then since $C(1:s) = \left(\begin{smallmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{smallmatrix}\right)$, we can partition C into the following form:

$$C = \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{pmatrix}, \tag{4}$$

where $C_{11} \in \mathbb{R}^{(n-s)\times(n-s)}$, $C_{22} \in \mathbb{R}^{(2s-n)\times(2s-n)}$, $C_{33} \in \mathbb{R}^{(n-s)\times(n-s)}$. By (4) and comparing the two sides of $C = E_nCE_n$, we obtain $C_{13} = E_{n-s}C_{31}E_{n-s}$, $C_{22} = E_{2s-n}C_{22}E_{2s-n}$, $C_{23} = E_{2s-n}C_{21}E_{n-s}$, $C_{32} = E_{n-s}C_{12}E_{2s-n}$, and $C_{33} = E_{n-s}C_{11}E_{n-s}$. Let $H = C_{13}E_{n-s}$; we get $C_{13} = HE_{n-s}$ and $C_{31} = E_{n-s}H$. Substituting C_{13}, C_{23}, C_{31}, C_{32}, and C_{33} into (4) and noticing that $C_{22} = E_{2s-n}C_{22}E_{2s-n}$, we have (3). □

In order to investigate the solvability of Problem I, we need the following lemmas.

LEMMA 3 (see [21, Lemma 1.3]). *Given $X, G \in \mathbb{R}^{n\times m}$ with $\operatorname{rank}(X) = l$. Then $YX = G$ has a solution $Y \in \mathbb{R}^{n\times n}$ if and only if $GX^+X = G$. In this case the general solution is*

$$Y = GX^+ + ZU_2^T,$$

where $U_2 \in \mathbb{R}^{n\times(n-l)}$, $U_2^TU_2 = I_{n-l}$, $\mathcal{R}(U_2) = \mathcal{N}(X^T)$, and $Z \in \mathbb{R}^{n\times(n-l)}$ is arbitrary.

LEMMA 4 (see [24, Lemma 3.1]). *Given any $E, F \in \mathbb{R}^{n\times n}$. Then there exists a unique $Y^* \in \mathbb{R}^{n\times n}$ such that*

$$\|Y^* - E\|^2 + \|Y^* - F\|^2 = \min_{Y\in\mathbb{R}^{n\times n}} \{\|Y - E\|^2 + \|Y - F\|^2\}$$

and

$$Y^* = \frac{E + F}{2}.$$

LEMMA 5. *Given any $E, F \in \mathbb{R}^{u\times v}$, $D = \operatorname{diag}(d_1, \ldots, d_v) > 0$ and $\Theta = \operatorname{diag}(\theta_1, \ldots, \theta_v)$, where $\theta_i = 1/(1 + d_i^2)$. Then there exists a unique $Y^* \in \mathbb{R}^{u\times v}$ such that*

$$\|Y^* - E\|^2 + \|Y^*D - F\|^2 = \min_{Y\in\mathbb{R}^{u\times v}} \{\|Y - E\|^2 + \|YD - F\|^2\}$$

and

$$Y^* = (E + FD)\Theta.$$

Proof. Let $Y = (y_{ij}), E = (e_{ij}), F = (f_{ij})$. Since

$$\begin{aligned}
\|Y - E\|^2 + \|YD - F\|^2 &= \sum_{i=1}^{u}\sum_{j=1}^{v}(y_{ij} - e_{ij})^2 + \sum_{i=1}^{u}\sum_{j=1}^{v}(y_{ij}d_j - f_{ij})^2 \\
&= \sum_{i=1}^{u}\sum_{j=1}^{v}[y_{ij}^2(1 + d_j^2) - 2y_{ij}(e_{ij} + f_{ij}d_j) + e_{ij}^2 + f_{ij}^2] \\
&= \sum_{i=1}^{u}\sum_{j=1}^{v}(1 + d_j^2)\left[\left(y_{ij} - \frac{e_{ij} + f_{ij}d_j}{1 + d_j^2}\right)^2 + \frac{e_{ij}^2 + f_{ij}^2}{1 + d_j^2} - \frac{(e_{ij} + f_{ij}d_j)^2}{(1 + d_j^2)^2}\right].
\end{aligned}$$

Thus there exists $Y \in \mathbb{R}^{u\times v}$ such that $\|Y - E\|^2 + \|YD - F\|^2 = \min$ is equivalent to $y_{ij} = (e_{ij} + f_{ij}d_j)/(1 + d_j^2)$; i.e., $Y^* = (E + FD)\Theta$. ∎

From Lemma 5, we can easily see that Lemma 4 is a special case of Lemma 5 where $u = v = n$, $D = I_n$, and $\Theta = 1/2I_n$.

3. Solvability conditions and general solutions of Problem I. Before we come to Problem I, we first note the following facts: For a real matrix $C \in \mathcal{C}_n$, its complex eigenvectors and eigenvalues are complex conjugate pairs. If $a \pm b\sqrt{-1}$ and $\mathbf{x} \pm \sqrt{-1}\mathbf{y}$ are one of its eigenpairs, then we have $C\mathbf{x} = a\mathbf{x} - b\mathbf{y}$ and $C\mathbf{y} = a\mathbf{y} + b\mathbf{x}$, i.e.,

$$C[\mathbf{x}, \mathbf{y}] = [\mathbf{x}, \mathbf{y}]\begin{pmatrix} a & b \\ -b & a \end{pmatrix}.$$

Therefore, without loss of generality, we can assume that $X \in \mathbb{R}^{n\times m}$ and

$$\Lambda = \operatorname{diag}(\Psi_1, \Psi_2, \dots, \Psi_l, g_1, \dots, g_{m-2l}) \in \mathbb{R}^{m\times m}, \tag{5}$$

where $\Psi_i = \left(\begin{smallmatrix} a_i & b_i \\ -b_i & a_i \end{smallmatrix}\right)$ with a_i, b_i, and g_i as real numbers.

THEOREM 1. *Given $X \in \mathbb{R}^{n\times m}$ and Λ as in (5), and $C_0 \in \mathbb{R}^{s\times s}$, where $s < n-k$. Partition $K^T X$ as*

$$K^T X = \begin{pmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{pmatrix}, \quad \tilde{X}_2 \in \mathbb{R}^{k\times m}. \tag{6}$$

Define

$$M_1 = [I_s, O_1]U_2, \quad M_2 = [I_s, O_2]V_2, \tag{7}$$

where $U_2 \in \mathbb{R}^{(n-k)\times(n-k-l_1)}$ and $V_2 \in \mathbb{R}^{k\times(k-l_2)}$ are column orthonormal, $\mathcal{R}(U_2) = \mathcal{N}(\tilde{X}_1^T)$, $\mathcal{R}(V_2) = \mathcal{N}(\tilde{X}_2^T)$, $l_1 = \operatorname{rank}(\tilde{X}_1)$, $l_2 = \operatorname{rank}(\tilde{X}_2)$, and $O_1 \in \mathbb{R}^{s\times(n-k-s)}$ and $O_2 \in \mathbb{R}^{s\times(k-s)}$ are zero matrices. Suppose that the generalized singular value decomposition (GSVD) of the matrix pair M_1^T and M_2^T is

$$M_1^T = P\Sigma_1 S^T, \quad M_2^T = Q\Sigma_2 S^T, \tag{8}$$

where S is an s-by-s nonsingular matrix, $P \in \mathbb{R}^{(n-k-l_1)\times(n-k-l_1)}$, $Q \in \mathbb{R}^{(k-l_2)\times(k-l_2)}$ are orthogonal, and

$$\Sigma_1 = \begin{array}{c} \\ r \\ t \\ h-r-t \\ s-h \end{array} \begin{array}{c} \begin{array}{cccc} r & t & h-r-t & s-h \end{array} \\ \begin{pmatrix} I_r & & & \\ & \Gamma_1 & & O \\ & & O_3 & \end{pmatrix} \end{array}, \quad \Sigma_2 = \begin{array}{c} \\ r \\ t \\ h-r-t \\ s-h \end{array} \begin{array}{c} \begin{array}{cccc} r & t & h-r-t & s-h \end{array} \\ \begin{pmatrix} O_4 & & & \\ & \Gamma_2 & & O \\ & & I_{h-r-t} & \end{pmatrix} \end{array} \tag{9}$$

with $h = \operatorname{rank}(M) = \operatorname{rank}([M_1, M_2])$, $r = h - \operatorname{rank}(M_2)$, $t = \operatorname{rank}(M_1) + \operatorname{rank}(M_2) - h$, O, O_3, and O_4 as zero matrices of size implied by context, and $\Gamma_1 = \operatorname{diag}(\gamma_1, \gamma_2, \dots, \gamma_t)$, $\Gamma_2 = \operatorname{diag}(\delta_1, \delta_2, \dots, \delta_t)$ with $1 \ge \gamma_t \ge \cdots \ge \gamma_1 > 0$, $0 < \delta_1 \le \cdots \le \delta_t$, $\gamma_i^2 + \delta_i^2 = 1$ for $i = 1, \dots, t$. Let

$$\tilde{G} = 2C_0 - [I_s, O_1]\tilde{X}_1\Lambda\tilde{X}_1^+[I_s, O_1]^T - [I_s, O_2]\tilde{X}_2\Lambda\tilde{X}_2^+[I_s, O_2]^T \tag{10}$$

and partition $\tilde{G}S^{-T}$ into the following form:

$$\tilde{G}S^{-T} = \begin{array}{c} \\ r \\ t \\ s-r-t \end{array} \begin{array}{c} \begin{array}{cccc} r & t & h-r-t & s-h \end{array} \\ \begin{pmatrix} G_{11} & G_{12} & G_{13} & G_{14} \\ G_{21} & G_{22} & G_{23} & G_{24} \\ G_{31} & G_{32} & G_{33} & G_{34} \end{pmatrix} \end{array}. \tag{11}$$

Then there exists a matrix $C \in \mathcal{C}_n$ such that $CX = X\Lambda$ and $C(1:s) = C_0$ if and only if

$$\tilde{X}_1\Lambda\tilde{X}_1^+\tilde{X}_1 = \tilde{X}_1\Lambda, \quad \tilde{X}_2\Lambda\tilde{X}_2^+\tilde{X}_2 = \tilde{X}_2\Lambda, \quad \text{and} \quad [G_{14}^T, G_{24}^T, G_{34}^T] = 0. \tag{12}$$

In this case, the general solution is given by

$$C = K\begin{pmatrix} \tilde{X}_1\Lambda\tilde{X}_1^+ + Z_1U_2^T & 0 \\ 0 & \tilde{X}_2\Lambda\tilde{X}_2^+ + Z_2V_2^T \end{pmatrix}K^T \tag{13}$$

with

$$Z_1 = \begin{array}{c} \\ r \\ t \\ s-r-t \\ n-k-s \end{array} \begin{array}{c} \begin{array}{ccc} r & t & n-k-l_1-r-t \end{array} \\ \begin{pmatrix} G_{11} & X_{12} & X_{13} \\ G_{21} & X_{22} & X_{23} \\ G_{31} & X_{32} & X_{33} \\ X_{41} & X_{42} & X_{43} \end{pmatrix} \end{array} P^T,$$

$$Z_2 = \begin{array}{c} \\ r \\ t \\ s-r-t \\ k-s \end{array} \begin{array}{c} \begin{array}{ccc} k-l_2+r-h & t & h-r-t \end{array} \\ \begin{pmatrix} Y_{11} & (G_{12}-X_{12}\Gamma_1)\Gamma_2^{-1} & G_{13} \\ Y_{21} & (G_{22}-X_{22}\Gamma_1)\Gamma_2^{-1} & G_{23} \\ Y_{31} & (G_{32}-X_{32}\Gamma_1)\Gamma_2^{-1} & G_{33} \\ Y_{41} & Y_{42} & Y_{43} \end{pmatrix} \end{array} Q^T,$$

where X_{12}, X_{13}, X_{22}, X_{23}, X_{32}, X_{33}, X_{41}, X_{42}, X_{43}, Y_{11}, Y_{21}, Y_{31}, Y_{41}, Y_{42}, *and* Y_{43} *are arbitrary matrices.*

Proof. By Lemmas 1 and 2, $C \in \mathcal{C}_n$ is a solution to Problem I if and only if there exist $F_1 \in \mathbb{R}^{(n-k)\times(n-k)}$ and $F_2 \in \mathbb{R}^{k\times k}$ such that

$$C = K \begin{pmatrix} F_1 & 0 \\ 0 & F_2 \end{pmatrix} K^T, \quad F_1(1:s) + F_2(1:s) = 2C_0 \tag{14}$$

and

$$K \begin{pmatrix} F_1 & 0 \\ 0 & F_2 \end{pmatrix} K^T X = X\Lambda. \tag{15}$$

Using (6), (15) is equivalent to

$$F_1\tilde{X}_1 = \tilde{X}_1\Lambda \quad \text{and} \quad F_2\tilde{X}_2 = \tilde{X}_2\Lambda. \tag{16}$$

According to Lemma 3, (16) have solutions if and only if

$$\tilde{X}_1\Lambda\tilde{X}_1^+\tilde{X}_1 = \tilde{X}_1\Lambda, \quad \tilde{X}_2\Lambda\tilde{X}_2^+\tilde{X}_2 = \tilde{X}_2\Lambda.$$

Moreover in this case, the general solution of (16) is given by

$$F_1 = \tilde{X}_1\Lambda\tilde{X}_1^+ + Z_1U_2^T, \tag{17}$$

$$F_2 = \tilde{X}_2\Lambda\tilde{X}_2^+ + Z_2V_2^T, \tag{18}$$

where $Z_1 \in \mathbb{R}^{(n-k)\times(n-k-l_1)}$ and $Z_2 \in \mathbb{R}^{k\times(k-l_2)}$ are both arbitrary. Putting (17) and (18) into $F_1(1:s) + F_2(1:s) = 2C_0$, and using the definition of $M_1, M_2, \tilde{G}$ and the GSVD of the matrix pair M_1^T and M_2^T, it is easy to show that Z_1 and Z_2 must satisfy

$$[I_s, O_1]Z_1P\Sigma_1 + [I_s, O_2]Z_2Q\Sigma_2 = \tilde{G}S^{-T}. \tag{19}$$

Partition Z_1P and Z_2Q into the following form:

$$Z_1P = \begin{pmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \\ X_{41} & X_{42} & X_{43} \end{pmatrix}, \quad Z_2Q = \begin{pmatrix} Y_{11} & Y_{12} & Y_{13} \\ Y_{21} & Y_{22} & Y_{23} \\ Y_{31} & Y_{32} & Y_{33} \\ Y_{41} & Y_{42} & Y_{43} \end{pmatrix}. \tag{20}$$

Substituting (9), (11), and (20) into (19) yields

$$\begin{pmatrix} X_{11} & X_{12}\Gamma_1 + Y_{12}\Gamma_2 & Y_{13} & 0 \\ X_{21} & X_{22}\Gamma_1 + Y_{22}\Gamma_2 & Y_{23} & 0 \\ X_{31} & X_{32}\Gamma_1 + Y_{32}\Gamma_2 & Y_{33} & 0 \end{pmatrix} = \begin{pmatrix} G_{11} & G_{12} & G_{13} & G_{14} \\ G_{21} & G_{22} & G_{23} & G_{24} \\ G_{31} & G_{32} & G_{33} & G_{34} \end{pmatrix}. \tag{21}$$

Thus (21), and hence (19) holds if and only if

$$[G_{14}^T, G_{24}^T, G_{34}^T] = 0, \tag{22}$$

$$X_{11} = G_{11}, \quad X_{21} = G_{21}, \quad X_{31} = G_{31}, \quad Y_{13} = G_{13}, \quad Y_{23} = G_{23}, \quad Y_{33} = G_{33}, \tag{23}$$

$$Y_{12} = (G_{12} - X_{12}\Gamma_1)\Gamma_2^{-1}, \quad Y_{22} = (G_{22} - X_{22}\Gamma_1)\Gamma_2^{-1}, \quad Y_{32} = (G_{32} - X_{32}\Gamma_1)\Gamma_2^{-1}. \tag{24}$$

Therefore, the solvability conditions for Problem I and the general expression of the solution of Problem I are obtained by (14), (16)–(18), (20), and (22)–(24). □

THEOREM 2. *Given $X \in \mathbb{R}^{n\times m}$ and Λ as in* (5)*, and $C_0 \in \mathbb{R}^{s\times s}$, where $s \geq n-k$, partition C_0 and X as*

$$C_0 = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}, \quad X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}, \tag{25}$$

where $C_{11} \in \mathbb{R}^{(n-s)\times(n-s)}$, $C_{22} \in \mathbb{R}^{(2s-n)\times(2s-n)}$, $X_1, X_3 \in \mathbb{R}^{(n-s)\times m}$ and $X_2 \in \mathbb{R}^{(2s-n)\times m}$. Let

$$U = [X_1, E_{n-s}X_3] \tag{26}$$

and

$$V = [E_{n-s}X_3\Lambda - C_{12}E_{2s-n}X_2 - C_{11}E_{n-s}X_3, X_1\Lambda - C_{11}X_1 - C_{12}X_2]. \tag{27}$$

Then Problem I *is solvable if and only if*

$$VU^+U = V, \quad C_{21}X_1 + C_{22}X_2 + E_{2s-n}C_{21}E_{n-s}X_3 = X_2\Lambda, \quad C_{22} \in \mathcal{C}_{2s-n}. \tag{28}$$

In this case, the general solution to Problem I *can be expressed as*

$$C = \begin{pmatrix} C_{11} & C_{12} & HE_{n-s} \\ C_{21} & C_{22} & E_{2s-n}C_{21}E_{n-s} \\ E_{n-s}H & E_{n-s}C_{12}E_{2s-n} & E_{n-s}C_{11}E_{n-s} \end{pmatrix}, \tag{29}$$

where $H = VU^+ + WQ_2^T$, where $Q_2 \in \mathbb{R}^{(n-s)\times(n-s-l_3)}$ is orthogonal, $\mathcal{R}(Q_2) = \mathcal{N}(U^T)$, $l_3 = \operatorname{rank}(U)$, and $W \in \mathbb{R}^{(n-s)\times(n-s-l_3)}$ is arbitrary.

Proof. By Lemma 2, there exists $C \in \mathcal{C}_n$ such that $CX = X\Lambda$ and $C_0 = C(1:s)$ if and only if there exists $H \in \mathbb{R}^{(n-s)\times(n-s)}$ such that

$$C = \begin{pmatrix} C_{11} & C_{12} & HE_{n-s} \\ C_{21} & C_{22} & E_{2s-n}C_{21}E_{n-s} \\ E_{n-s}H & E_{n-s}C_{12}E_{2s-n} & E_{n-s}C_{11}E_{n-s} \end{pmatrix}, \quad CX = X\Lambda.$$

Equivalently,

$$C_{21}X_1 + C_{22}X_2 + E_{2s-n}C_{21}E_{n-s}X_3 = X_2\Lambda, \quad C_{22} \in \mathcal{C}_{2s-n},$$

and

$$HU = V. \tag{30}$$

From Lemma 3, (30) holds if and only if

$$VU^+U = V, \tag{31}$$

and when (31) holds, H can be expressed as

$$H = VU^+ + WQ_2^T.$$

Thus Problem I is solvable if and only if the conditions in (28) hold, and the general solution can be expressed as (29). □

4. The solution of Problem II. In this section, we solve Problem II over $\mathcal{L}_S$ when $\mathcal{L}_S$ is nonempty.

THEOREM 3. *Given $X \in \mathbb{R}^{n\times m}$ and Λ as in* (5)*, and $C_0 \in \mathbb{R}^{s\times s}$, where $s < n-k$, suppose the solution set $\mathcal{L}_S$ of Problem* I *is nonempty. Let*

$$K^T\tilde{C}K = \begin{pmatrix} \tilde{C}_{11} & \tilde{C}_{12} \\ \tilde{C}_{21} & \tilde{C}_{22} \end{pmatrix}, \tag{32}$$

$$(\tilde{C}_{11} - \tilde{X}_1\Lambda\tilde{X}_1^+)U_2P = \begin{pmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \\ E_{41} & E_{42} & E_{43} \end{pmatrix}, \tag{33}$$

$$(\tilde{C}_{22} - \tilde{X}_2\Lambda\tilde{X}_2^+)V_2Q = \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \\ F_{41} & F_{42} & F_{43} \end{pmatrix}, \tag{34}$$

where $\tilde{X}_1, \tilde{X}_2$ are the same as (6)*, the size of matrices $\tilde{C}_{11}$ and $\tilde{C}_{22}$ is the same as F_1 and F_2 in* (14)*, respectively, and the partition form of* (33) *and* (34) *is the same as* (20)*. Then Problem* II *has a unique solution and the solution is given by*

$$C^* = K\begin{pmatrix} \tilde{X}_1\Lambda\tilde{X}_1^+ + Z_1U_2^T & 0 \\ 0 & \tilde{X}_2\Lambda\tilde{X}_2^+ + Z_2V_2^T \end{pmatrix}K^T, \tag{35}$$

where

$$Z_1 = \begin{pmatrix} G_{11} & \hat{X}_{12} & E_{13} \\ G_{21} & \hat{X}_{22} & E_{23} \\ G_{31} & \hat{X}_{32} & E_{33} \\ E_{41} & E_{42} & E_{43} \end{pmatrix}P^T, \quad Z_2 = \begin{pmatrix} F_{11} & (G_{12} - \hat{X}_{12}\Gamma_1)\Gamma_2^{-1} & G_{13} \\ F_{21} & (G_{22} - \hat{X}_{22}\Gamma_1)\Gamma_2^{-1} & G_{23} \\ F_{31} & (G_{32} - \hat{X}_{32}\Gamma_1)\Gamma_2^{-1} & G_{33} \\ F_{41} & F_{42} & F_{43} \end{pmatrix}Q^T.$$

$$\hat{X}_{12} = (G_{12}\Gamma_1\Gamma_2^{-2} + E_{12} - F_{12}\Gamma_1\Gamma_2^{-1})\Theta,$$

$$\hat{X}_{22} = (G_{22}\Gamma_1\Gamma_2^{-2} + E_{22} - F_{22}\Gamma_1\Gamma_2^{-1})\Theta,$$

$$\hat{X}_{32} = (G_{32}\Gamma_1\Gamma_2^{-2} + E_{32} - F_{32}\Gamma_1\Gamma_2^{-1})\Theta,$$

$$\Theta = \mathrm{diag}(\theta_1, \ldots, \theta_t), \quad \theta_i = \frac{\delta_i^2}{\delta_i^2 + \gamma_i^2}.$$

Proof. When $\mathcal{L}_S$ is nonempty, it is easy to verify from (13) that $\mathcal{L}_S$ is a closed convex set. Since $\mathbb{R}^{n\times n}$ is a uniformly convex Banach space under the Frobenius norm, there exists a unique solution for Problem II [6, p. 22]. Moreover, because the Frobenius norm is unitary invariant, Problem II is equivalent to

$$\min_{C\in\mathcal{L}_S} \|K^T\tilde{C}K - K^TCK\|^2. \tag{36}$$

By (13) and (32)–(34), (36) is equivalent to

$$\min_{Z_1\in\mathbb{R}^{(n-k)\times(n-k-l_1)}} \|\tilde{X}_1\Lambda\tilde{X}_1^+ + Z_1U_2^T - \tilde{C}_{11}\|^2 + \min_{Z_2\in\mathbb{R}^{k\times(k-l_2)}} \|\tilde{X}_2\Lambda\tilde{X}_2^+ + Z_2V_2^T - \tilde{C}_{22}\|^2.$$

Equivalently,

$$\min_{Z_1\in\mathbb{R}^{(n-k)\times(n-k-l_1)}} \|Z_1 - (\tilde{C}_{11} - \tilde{X}_1\Lambda\tilde{X}_1^+)U_2\|^2 + \min_{Z_2\in\mathbb{R}^{k\times(k-l_2)}} \|Z_2 - (\tilde{C}_{22} - \tilde{X}_2\Lambda\tilde{X}_2^+)V_2\|^2.$$

Clearly, the solution is given by X_{12}, X_{13}, X_{22}, X_{23}, X_{32}, X_{33}, X_{41}, X_{42}, X_{43} and Y_{11}, Y_{21}, Y_{31}, Y_{41}, Y_{42}, Y_{43} such that

$$\|X_{13} - E_{13}\| = \min, \quad \|X_{23} - E_{23}\| = \min, \quad \|X_{33} - E_{33}\| = \min, \tag{37}$$

$$\|X_{41} - E_{41}\| = \min, \quad \|X_{42} - E_{42}\| = \min, \quad \|X_{43} - E_{43}\| = \min, \tag{38}$$

$$\|Y_{11} - F_{11}\| = \min, \quad \|Y_{21} - F_{21}\| = \min, \quad \|Y_{31} - F_{31}\| = \min, \tag{39}$$

$$\|Y_{41} - F_{41}\| = \min, \quad \|Y_{42} - F_{42}\| = \min, \quad \|Y_{43} - F_{43}\| = \min, \tag{40}$$

$$\|X_{12} - E_{12}\|^2 + \|X_{12}\Gamma_1\Gamma_2^{-1} - (G_{12}\Gamma_2^{-1} - F_{12})\|^2 = \min, \tag{41}$$

$$\|X_{22} - E_{22}\|^2 + \|X_{22}\Gamma_1\Gamma_2^{-1} - (G_{22}\Gamma_2^{-1} - F_{22})\|^2 = \min, \tag{42}$$

$$\|X_{32} - E_{32}\|^2 + \|X_{32}\Gamma_1\Gamma_2^{-1} - (G_{32}\Gamma_2^{-1} - F_{32})\|^2 = \min\,. \tag{43}$$

By (37)–(40), we get

$$X_{13} = E_{13}, \quad X_{23} = E_{23}, \quad X_{33} = E_{33}, \quad X_{41} = E_{41}, \quad X_{42} = E_{42}, \quad X_{43} = E_{43}, \tag{44}$$

$$Y_{11} = F_{11}, \quad Y_{21} = F_{21}, \quad Y_{31} = F_{31}, \quad Y_{41} = F_{41}, \quad Y_{42} = F_{42}, \quad Y_{42} = F_{42}, \quad Y_{43} = F_{43}. \tag{45}$$

Applying Lemma 5 to (41)–(43), we obtain

(46)
$$X_{12} = (G_{12}\Gamma_1\Gamma_2^{-2} + E_{12} - F_{12}\Gamma_1\Gamma_2^{-1})\Theta, \quad X_{22} = (G_{22}\Gamma_1\Gamma_2^{-2} + E_{22} - F_{22}\Gamma_1\Gamma_2^{-1})\Theta,$$

and

$$X_{32} = (G_{32}\Gamma_1\Gamma_2^{-2} + E_{32} - F_{32}\Gamma_1\Gamma_2^{-1})\Theta. \tag{47}$$

By (13) and (44)–(47), we have the unique solution of Problem II given by (35). ∎

THEOREM 4. *Given $X \in \mathbb{R}^{n\times m}$ and Λ as in* (5)*, and $C_0 \in \mathbb{R}^{s\times s}$, where $s \geq n-k$. Suppose the solution set $\mathcal{L}_S$ of Problem* I *is nonempty. Let*

$$\tilde{C} = \begin{pmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ W_{31} & W_{32} & W_{33} \end{pmatrix}, \tag{48}$$

where $W_{11} \in \mathbb{R}^{(n-s)\times(n-s)}$, $W_{22} \in \mathbb{R}^{(2s-n)\times(2s-n)}$, and $W_{33} \in \mathbb{R}^{(n-s)\times(n-s)}$. Then Problem II *has a unique solution which can be expressed as*

$$C^* = \begin{pmatrix} C_{11} & C_{12} & \hat{H}E_{n-s} \\ C_{21} & C_{22} & E_{2s-n}C_{21}E_{n-s} \\ E_{n-s}\hat{H} & E_{n-s}C_{12}E_{2s-n} & E_{n-s}C_{11}E_{n-s} \end{pmatrix}, \tag{49}$$

where

$$\hat{H} = VU^+ + \hat{W}Q_2^T, \quad \hat{W} = \frac{1}{2}(W_{13}E_{n-s} + E_{n-s}W_{31})Q_2.$$

Proof. As in the proof of Theorem 3, we can show that Problem II has a unique solution in $\mathcal{L}_S$. By (29) and (48), we know that Problem II is equivalent to

$$\min_{H\in\mathbb{R}^{(n-s)\times(n-s)}} (\|HE_{n-s} - W_{13}\|^2 + \|E_{n-s}H - W_{31}\|^2).$$

Equivalently,

$$\min_{H\in\mathbb{R}^{(n-s)\times(n-s)}} (\|H - W_{13}E_{n-s}\|^2 + \|H - E_{n-s}W_{31}\|^2).$$

By Lemma 4, it is in turn equivalent to

$$\min_{H\in\mathbb{R}^{(n-s)\times(n-s)}} \left\| H - \frac{1}{2}(W_{13}E_{n-s} + E_{n-s}W_{31}) \right\|.$$

That is,

$$\min_{W\in\mathbb{R}^{(n-s)\times(n-k-l_3)}} \|VU^+ + WQ_2^T - \frac{1}{2}(W_{13}E_{n-s} + E_{n-s}W_{31})\|.$$

Since Q_2 is orthogonal and $U^+Q_2 = 0$, we have

$$W = \frac{1}{2}(W_{13}E_{n-s} + E_{n-s}W_{31})Q_2.$$

Therefore, the solution of Problem II can be expressed as (49). ∎

Based on the above discussion, we give the following algorithm for solving Problem II.

ALGORITHM I.

Given $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m] \in \mathbb{R}^{n\times m}$ and Λ as in (5), $C_0 \in \mathbb{R}^{s\times s}$, and $\tilde{C} \in \mathbb{R}^{n\times n}$.

1. Calculate $k = [n/2]$.
2. If $s < n - k$, then
   (a) Compute $\tilde{X}_1$ and $\tilde{X}_2$ by (6) and then compute $\tilde{X}_1^+$ and $\tilde{X}_2^+$.
   (b) If $\tilde{X}_1\Lambda\tilde{X}_1^+\tilde{X}_1 = \tilde{X}_1\Lambda$ and $\tilde{X}_2\Lambda\tilde{X}_2^+\tilde{X}_2 = \tilde{X}_2\Lambda$, then we continue. Otherwise we stop.
   (c) Calculate M_1 and M_2 as in (7).
   (d) Construct the GSVD of the matrix pair $[M_1^T, M_2^T]$ by (8).
   (e) Compute $\tilde{G}$ as in (10) and then calculate $\tilde{G}S^{-T}$.
   (f) Partition $\tilde{G}S^{-T}$ as in (11).
   (g) If G_{14}, G_{24}, and G_{34} are zero matrices, then calculate C^* as in (35). Otherwise we stop.
3. Else
   (a) Partition X and C_0 as in (25), and calculate U and V as in (26) and (27).
   (b) If the conditions of (28) are satisfied, then compute C^* as in (49). Otherwise we stop.

Now we consider the computational complexity of our algorithm. We first consider the cost of step 2. For substep (a), since K has only 2 nonzero entries per row, it requires $O(nm)$ operations to compute $\tilde{X}_1$ and $\tilde{X}_2$. Then using singular value decomposition (SVD) to compute $\tilde{X}_1^+$ and $\tilde{X}_2^+$ requires $O(n^2m + m^3)$ operations. Substep (b) obviously requires $O(n^2m)$ operations. For substep (c), because U_2 and V_2 can be obtained by SVD of $\tilde{X}_1$ and $\tilde{X}_2$ in substep (a), respectively, it requires no operations to compute M_1 and M_2. For substep (d), if we use Paige's algorithm [19] to compute the GSVD of the matrix pair $[M_1^T, M_2^T]$, then the cost will be of $O(s^2(n-l_1-l_2-s/3))$ operations if $n-l_1-l_2 \geq s$ ($O((n-l_1-l_2)^2(s-(n-l_1-l_2)/3))$ operations if $n-l_1-l_2 \leq s$). Substep (e) requires $O(n^2m+s^3)$ operations. Substep (f) requires no operations. Finally, because of the sparsity of K again, step (g) requires $O(n^2(n-k-l_1)+n(n-k-l_1)^2+n^2(k-l_2)+n(k-l_2)^2)$ operations. Thus the total complexity of step 2 is $O(n^2(n-l_1-l_2)+s^2(n-l_1-l_2-s/3)+s^3+n^2m+m^3)$ if $n-l_1-l_2 \geq s$ ($O(n^2(n-l_1-l_2)+s^2(s-(n-l_1-l_2)/3)+s^3+n^2m+m^3)$ if $n-l_1-l_2 \leq s$).

Next, we consider the cost of step 3. For substep (a), since E_n is a backward identity matrix, it requires $O((n-s)^2m+(n-s)(2s-n)m)$ operations to form U and V. For substep (b), using SVD to compute U^+ requires $O((n-s)^2m+m^3)$ operations. If we compute VU^+U as $[V(U^+U)]$, then the cost will only be of $O(m^2(n-s))$ operations. Thus the cost for substep (b) is $O((n-s)^2m+m^3+m^2(n-s)+(n-s)^3)$. Therefore, the total cost of step 3 is $O((n-s)^3+(n-s)^2m+(n-s)(2s-n)m+m^2(n-s)+m^3)$.

From above, we know that the total cost of the algorithm is the cost required by step 2 if $s < n-k$ or by step 3 if $s \geq n-k$. We remark that in practice $m \ll n$.

5. Numerical experiments. In this section, we will demonstrate the algorithm using Matlab.

Example 1. We consider the following Hopfield neural network system:

$$\frac{d\mathbf{u}}{dt} = T^{-1}(-\mathbf{u} + \Omega\mathbf{f}(\mathbf{u})), \tag{50}$$

where $T = \mathrm{diag}(\tau_1, \ldots, \tau_n)$, $\Omega = [\omega_{ij}]$, and $\mathbf{f} = [f_1(u_1), \ldots, f_n(u_n)]^T$ with $f_i(u_i)$ as squashing functions; see [8] for details.

In this example, we design a neural network such that $\mathbf{u}^*$ is a stable equilibrium,

with $f_i(u_i^*) = 1/(1+e^{-u_i^*}) \neq 0$. It is known that $\mathbf{u}^*$ is an equilibrium only if

$$\mathbf{u}^* = \Omega \mathbf{f}. \tag{51}$$

It implies that

$$\Omega = TCG_d^{-1} + G_d^{-1}, \tag{52}$$

where C satisfies that

$$CG_d^{-1}\mathbf{f} = T^{-1}(\mathbf{u}^* - G_d^{-1}\mathbf{f}). \tag{53}$$

Here, $G_d = \mathrm{diag}(f_1^{(1)}(u_1^*), \ldots, f_n^{(1)}(u_n^*))$, where $(\cdot)^{(1)}$ denotes the 1st derivatives.

For any given T, the design problem is reduced to finding a stable matrix C that maps $G_d^{-1}\mathbf{f}$ to $T^{-1}(\mathbf{u}^* - G_d^{-1}\mathbf{f})$. Moreover, we know that if $T^{-1}(\mathbf{u}^* - G_d^{-1}\mathbf{f}) = \lambda G_d^{-1}\mathbf{f}$ for some real negative number λ, then there exists a stable matrix C such that (53) holds; see [8, Theorem 4.1].

In practice, we may be interested that the matrix C is a centrosymmetric matrix and its s-by-s leading principal submatrix is the given matrix C_0. Moreover, we can obtain an experimental matrix $\tilde{C}$ which may not satisfy the structural requirement (centrosymmetric or the submatrix constraint) and/or spectral requirement (having eigenpairs $G_d^{-1}\mathbf{f}$ and λ). We want to find such structural stable matrix C^* which maps $G_d^{-1}\mathbf{f}$ to $T^{-1}(\mathbf{u}^* - G_d^{-1}\mathbf{f}) = \lambda G_d^{-1}\mathbf{f}$ ($\lambda < 0$) and is the best approximation of $\tilde{C}$ in Frobenius norm. Therefore the design problem turns into Problems I and II proposed in this paper.

For demonstration, we let $n = 8$, $m = 1$, $s = 5$ and are given $\mathbf{u}^* = \mathbf{0}$. Then we have $f_i(u_i^*) = 1/2$ and $f_i^{(1)}(u_i^*) = 1/4$ for $i = 1, \ldots, n$. Thus $G_d = 1/4 I_n$ and $\mathbf{f} = 1/2\mathbf{e}$, where $\mathbf{e}$ denotes the n-vector of all ones. Therefore, the given eigenvector $G_d^{-1}\mathbf{f} = 2\mathbf{e}$. For this example, we chose $T = 0.4938 I_n$ so that one eigenvalue of C is $\lambda = -1/0.4938 = -2.0251$.

Given $X = G_d^{-1}\mathbf{f} = 2\mathbf{e}$, $\Lambda = \lambda = -2.0251$, and

$$C_0 = \begin{pmatrix} 1.0134 & -0.6262 & -0.6091 & 0.2024 & 0.8464 \\ 0.3118 & 0.1653 & 1.1857 & 0.8940 & 0.0265 \\ 0.1912 & 0.6515 & -0.9667 & 1.0504 & -0.5886 \\ -0.7399 & 0.4515 & -0.6165 & -0.5674 & -0.9952 \\ -0.0169 & -0.8830 & -0.2698 & -0.9952 & -0.5674 \end{pmatrix}.$$

Assume that from the experiment we get the following matrix $\tilde{C} \notin \mathcal{C}_8$:

$$\tilde{C} = \begin{pmatrix} 3.6448 & -1.5739 & 0.5661 & 1.2763 & 0.5473 & 0.5312 & 0.2992 & -1.2917 \\ 1.5866 & 0.1344 & 0.4095 & 1.1794 & -0.9925 & 0.8905 & 0.5602 & -1.1477 \\ 0.7641 & 0.6437 & -2.0927 & 1.5228 & 0.0533 & 0.8970 & 0.1428 & 0.5543 \\ -1.0982 & 1.4538 & -2.1948 & -1.4674 & -0.7619 & 0.1669 & 0.1910 & -1.4562 \\ 0.7249 & -1.8998 & -0.1476 & -0.7729 & 0.5174 & -2.3614 & -0.3332 & -0.3404 \\ 0.1476 & 0.8403 & -0.3028 & -0.4868 & 0.8683 & 0.4873 & -0.0583 & 1.8999 \\ 2.2642 & 1.8592 & 1.4312 & 0.6824 & 0.5707 & 1.9692 & 1.3696 & -0.6353 \\ -0.0637 & -0.4936 & 1.9980 & 1.9972 & -0.1334 & 0.8525 & -3.0381 & 0.5415 \end{pmatrix}.$$

We can show that Problem I is solvable. Then following the steps in the algorithm in section 4, we obtain the required matrix $C^* \in \mathcal{L}_S$ as follows:

$$C^* = \begin{pmatrix} 1.0134 & -0.6262 & -0.6091 & 0.2024 & 0.8464 & 0.1507 & -1.2111 & -1.7916 \\ 0.3118 & 0.1653 & 1.1857 & 0.8940 & 0.0265 & -1.3515 & -1.3027 & -1.9541 \\ 0.1912 & 0.6515 & -0.9667 & 1.0504 & -0.5886 & -0.8704 & -0.6760 & -0.8166 \\ -0.7399 & 0.4515 & -0.6165 & -0.5674 & -0.9952 & -0.2698 & -0.8830 & -0.0169 \\ -0.0169 & -0.8830 & -0.2698 & -0.9952 & -0.5674 & -0.6165 & 0.4515 & -0.7399 \\ -0.8166 & -0.6760 & -0.8704 & -0.5886 & 1.0504 & -0.9667 & 0.6515 & 0.1912 \\ -1.9541 & -1.3027 & -1.3515 & 0.0265 & 0.8940 & 1.1857 & 0.1653 & 0.3118 \\ -1.7916 & -1.2111 & 0.1507 & 0.8464 & 0.2024 & -0.6091 & -0.6262 & 1.0134 \end{pmatrix},$$

which satisfies $\|C^* - \tilde{C}\| = \min_{C \in \mathcal{L}_S} \|C - \tilde{C}\|$. Finally, the following matrix $\Omega^* = TC^*G_d^{-1} + G_d^{-1}$ can be calculated:

$$\Omega^* = \begin{pmatrix} 6.0016 & -1.2368 & -1.2031 & 0.3997 & 1.6719 & 0.2976 & -2.3922 & -3.5388 \\ 0.6158 & 4.3265 & 2.3420 & 1.7659 & 0.0523 & -2.6695 & -2.5731 & -3.8598 \\ 0.3776 & 1.2868 & 2.0907 & 2.0748 & -1.1625 & -1.7192 & -1.3352 & -1.6129 \\ -1.4615 & 0.8918 & -1.2176 & 2.8793 & -1.9657 & -0.5329 & -1.7441 & -0.0333 \\ -0.0333 & -1.7441 & -0.5329 & -1.9657 & 2.8793 & -1.2176 & 0.8918 & -1.4615 \\ -1.6129 & -1.3352 & -1.7192 & -1.1625 & 2.0748 & 2.0907 & 1.2868 & 0.3776 \\ -3.8598 & -2.5731 & -2.6695 & 0.0523 & 1.7659 & 2.3420 & 4.3265 & 0.6158 \\ -3.5388 & -2.3922 & 0.2976 & 1.6719 & 0.3997 & -1.2031 & -1.2368 & 6.0016 \end{pmatrix}.$$

Example 2. In this example, we demonstrate our algorithm in another way. For simplicity, we consider $n = 10$, $m = 3$, $s = 6$. We first choose a random matrix $\hat{C} \in \mathcal{C}_{10}$:

$$\hat{C} = \begin{pmatrix} 1.6405 & -0.1078 & -0.8875 & 0.3703 & -0.2894 & -0.6384 & 0.7080 & 0.2080 & 0.3988 & 0.8062 \\ -0.4574 & -0.8891 & -0.1455 & -0.0858 & -0.2658 & -1.3510 & 0.7036 & -0.3054 & 0.4304 & 1.4557 \\ 0.1118 & -0.1969 & 0.1812 & -0.2555 & 1.1810 & 0.5378 & 0.4137 & 0.8233 & -1.2063 & 1.3373 \\ -0.7977 & -0.0109 & 0.3346 & -0.3387 & 0.3376 & 0.2088 & -0.0052 & 0.0533 & 0.8645 & -0.2588 \\ 0.1512 & -0.5887 & -0.3039 & -0.0137 & 0.4058 & 0.1813 & 0.5433 & -0.1110 & 0.4449 & -0.0643 \\ -0.0643 & 0.4449 & -0.1110 & 0.5433 & 0.1813 & 0.4058 & -0.0137 & -0.3039 & -0.5887 & 0.1512 \\ -0.2588 & 0.8645 & 0.0533 & -0.0052 & 0.2088 & 0.3376 & -0.3387 & 0.3346 & -0.0109 & -0.7977 \\ 1.3373 & -1.2063 & 0.8233 & 0.4137 & 0.5378 & 1.1810 & -0.2555 & 0.1812 & -0.1969 & 0.1118 \\ 1.4557 & 0.4304 & -0.3054 & 0.7036 & -1.3510 & -0.2658 & -0.0858 & -0.1455 & -0.8891 & -0.4574 \\ 0.8062 & 0.3988 & 0.2080 & 0.7080 & -0.6384 & -0.2894 & 0.3703 & -0.8875 & -0.1078 & 1.6405 \end{pmatrix}.$$

Then we compute its eigenpairs: Three of the eigenvalues of $\hat{C}$ are 2.1176, $1.0359 \pm 1.1570\sqrt{-1}$. Let $\mathbf{x}_1$, $\mathbf{x}_2 \pm \sqrt{-1}\mathbf{x}_3$ be the corresponding eigenvectors. We now take

$$X = [\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_1] = \begin{pmatrix} -0.0659 & -0.2562 & -0.5799 \\ 0.0678 & 0.0191 & -0.2116 \\ -0.6079 & 0 & -0.2835 \\ 0.0422 & 0.0986 & 0.1571 \\ 0.0959 & -0.1867 & 0.1181 \\ 0.0959 & -0.1867 & 0.1181 \\ 0.0422 & 0.0986 & 0.1571 \\ -0.6079 & 0.0000 & -0.2835 \\ 0.0678 & 0.0191 & -0.2116 \\ -0.0659 & -0.2562 & -0.5799 \end{pmatrix}$$

and

$$\Lambda = \begin{pmatrix} 1.0359 & 1.1570 & 0 \\ -1.1570 & 1.0359 & 0 \\ 0 & 0 & 2.1176 \end{pmatrix}.$$

Given such X, Λ, and

$$C_0 = \begin{pmatrix} 1.6405 & -0.1078 & -0.8875 & 0.3703 & -0.2894 & -0.6384 \\ -0.4574 & -0.8891 & -0.1455 & -0.0858 & -0.2658 & -1.3510 \\ 0.1118 & -0.1969 & 0.1812 & -0.2555 & 1.1810 & 0.5378 \\ -0.7977 & -0.0109 & 0.3346 & -0.3387 & 0.3376 & 0.2088 \\ 0.1512 & -0.5887 & -0.3039 & -0.0137 & 0.4058 & 0.1813 \\ -0.0643 & 0.4449 & -0.1110 & 0.5433 & 0.1813 & 0.4058 \end{pmatrix},$$

we can verify that Problem I is solvable. Hence $\mathcal{L}_S$ is nonempty. We now perturb $\hat{C}$ by a random matrix to obtain a matrix $\tilde{C} \notin \mathcal{C}_{10}$:

$$\tilde{C} = \begin{pmatrix} 1.6510 & -0.0907 & -0.8789 & 0.3653 & -0.2906 & -0.6402 & 0.7090 & 0.2027 & 0.4049 & 0.8028 \\ -0.4538 & -0.8976 & -0.1401 & -0.0693 & -0.2665 & -1.3369 & 0.6919 & -0.3034 & 0.4402 & 1.4466 \\ 0.1097 & -0.1948 & 0.1928 & -0.2422 & 1.1870 & 0.5320 & 0.4281 & 0.8313 & -1.2258 & 1.3349 \\ -0.7985 & -0.0205 & 0.3434 & -0.3477 & 0.3422 & 0.2029 & 0.0032 & 0.0747 & 0.8667 & -0.2572 \\ 0.1457 & -0.5941 & -0.2902 & -0.0137 & 0.4016 & 0.1786 & 0.5295 & -0.1168 & 0.4491 & -0.0602 \\ -0.0578 & 0.4532 & -0.1160 & 0.5324 & 0.1973 & 0.4021 & -0.0297 & -0.3002 & -0.5844 & 0.1455 \\ -0.2607 & 0.8667 & 0.0397 & -0.0128 & 0.1987 & 0.3351 & -0.3423 & 0.3373 & -0.0018 & -0.7974 \\ 1.3200 & -1.2080 & 0.8347 & 0.4064 & 0.5380 & 1.1847 & -0.2747 & 0.1803 & -0.1985 & 0.1151 \\ 1.4712 & 0.4299 & -0.2972 & 0.7098 & -1.3340 & -0.2750 & -0.0797 & -0.1331 & -0.9039 & -0.4427 \\ 0.8031 & 0.3804 & 0.2401 & 0.7207 & -0.6404 & -0.2924 & 0.3609 & -0.8630 & -0.1050 & 1.6367 \end{pmatrix}.$$

Using the proposed algorithm in section 4, we can obtain $C^* \in \mathcal{L}_S$ such that $\|C^* - \tilde{C}\| = \min_{C \in \mathcal{L}_S} \|C - \tilde{C}\|$. Moreover, the solution C^* is given by

$$C^* = \begin{pmatrix} 1.6405 & -0.1078 & -0.8875 & 0.3703 & -0.2894 & -0.6384 & 0.7141 & 0.2080 & 0.3972 & 0.8084 \\ -0.4574 & -0.8891 & -0.1455 & -0.0858 & -0.2658 & -1.3510 & 0.7013 & -0.3054 & 0.4310 & 1.4549 \\ 0.1118 & -0.1969 & 0.1812 & -0.2555 & 1.1810 & 0.5378 & 0.4160 & 0.8233 & -1.2068 & 1.3381 \\ -0.7977 & -0.0109 & 0.3346 & -0.3387 & 0.3376 & 0.2088 & -0.0054 & 0.0533 & 0.8646 & -0.2588 \\ 0.1512 & -0.5887 & -0.3039 & -0.0137 & 0.4058 & 0.1813 & 0.5433 & -0.1110 & 0.4449 & -0.0643 \\ -0.0643 & 0.4449 & -0.1110 & 0.5433 & 0.1813 & 0.4058 & -0.0137 & -0.3039 & -0.5887 & 0.1512 \\ -0.2588 & 0.8646 & 0.0533 & -0.0054 & 0.2088 & 0.3376 & -0.3387 & 0.3346 & -0.0109 & -0.7977 \\ 1.3381 & -1.2068 & 0.8233 & 0.4160 & 0.5378 & 1.1810 & -0.2555 & 0.1812 & -0.1969 & 0.1118 \\ 1.4549 & 0.4310 & -0.3054 & 0.7013 & -1.3510 & -0.2658 & -0.0858 & -0.1455 & -0.8891 & -0.4574 \\ 0.8084 & 0.3972 & 0.2080 & 0.7141 & -0.6384 & -0.2894 & 0.3703 & -0.8875 & -0.1078 & 1.6405 \end{pmatrix}.$$

In addition, we note that if in Problem I, we also assume that the required matrix C is symmetric, i.e., C is bisymmetric, then Problem I is reduced to the inverse problem for submatrix constrained bisymmetric matrices discussed in [18]. For the corresponding solvability conditions, the algorithm for finding the best approximation solution to the corresponding best approximation problem and the numerical examples, we can refer to [18].

These examples and many other numerical experiments with the algorithm proposed in section 4 confirm our theoretical results in this paper.

6. Conclusions. In this paper, we discussed the inverse eigenproblem for the submatrix constrained centrosymmetric matrices. We also considered the best approximation solution in the corresponding solution set for the constrained inverse problem to a given matrix in Frobenius norm. The solvability conditions and the explicit formula for the solution are provided. We proposed an algorithm for finding the best approximation solution. Some tests are also given to illustrate our results.

Acknowledgments. The author is very grateful to Professor Raymond H. Chan for many helpful conversations. He also thanks the referees for their valuable comments and suggestions.

REFERENCES

[1] A. Andrew, *Eigenvectors of certain matrices*, Linear Algebra Appl., 7 (1973), pp. 151–162.

[2] Z. Bai and R. Chan, *Inverse eigenproblem for centrosymmetric and centroskew matrices and their approximation*, Theoret. Comput. Sci., 315 (2004), pp. 309–318.

[3] T. Chan, *An optimal circulant preconditioner for Toeplitz systems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 766–771.

[4] W. Chen, X. Wang, and T. Zhong, *The structure of weighting coefficient matrices of harmonic differential quadrature and its applications*, Commun. Numer. Methods Engrg., 12 (1996), pp. 455–460.

[5] X. Chen, *Theoretical Method and Its Application of Designing of Structured Dynamics in Machine*, The Mechanical Industry (Chinese) Press, Beijing, China, 1997, pp. 165–202.

[6] E. Cheney, *Introduction to Approximation Theory*, McGraw-Hill, New York, 1966.

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

[8] K. Chu and N. Li, *Designing the Hopfield neural network via pole assignment*, Internat. J. Systems Sci., 25 (1994), pp. 669–681.

[9] A. Collar, *On centrosymmetric and centroskew matrices*, Quart. J. Mech. Appl. Math., 15 (1962), pp. 265–281.

[10] L. Datta and S. Morgera, *Some results on matrix symmetries and a pattern recognition application*, IEEE Trans. Signal Process., 34 (1986), pp. 992–994.

[11] L. Datta and S. Morgera, *On the reducibility of centrosymmetric matrices—applications in engineering problems*, Circuits Systems Signal Process., 8 (1989), pp. 71–96.

[12] J. Delmas, *On Adaptive EVD asymptotic distribution of centro-symmetric covariance matrices*, IEEE Trans. Signal Process., 47 (1999), pp. 1402–1406.

[13] P. Delsarte and Y. Genin, *Spectral properties of finite Toeplitz matrices*, in Proceedings of the 1983 International Symposium on Mathematical Theory of Networks and Systems, Beer Sheva, Israel, 1983, Springer-Verlag, London, 1984, pp. 194–213.

[14] S. Friedland, *The reconstruction of a symmetric matrix from the spectral data*, J. Math. Anal. Appl., 71 (1979), pp. 412–422.

[15] T. Kailath and A. Sayed, *Fast Reliable Algorithms for Matrices with Structures*, SIAM, Philadelphia, 1999.

[16] N. Li, *A matrix inverse eigenvalue problem and its application*, Linear Algebra Appl., 266 (1997), pp. 143–152.

[17] T. Meng, *Experimental design and decision support*, in Expert Systems, The Technology of Knowledge Management and Decision Making for the 21st Century, Vol. 1, C. Leondes, ed., Academic Press, New York, 2001.

[18] Z. Peng, X. Hu, and L. Zhang, *The inverse problem of bisymmetric matrices with a submatrix constraint*, Numer. Linear Algebra Appl., 11 (2003), pp. 59–73.

[19] C. Paige, *Computing the generalized singular value decomposition*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1126–1146.

[20] V. Papanicolaou and D. Kravvaritis, *An inverse spectral problem for the Euler–Bernoulli equation for the vibrating beam*, Inverse Problems, 13 (1997), pp. 1083–1092.

[21] J. Sun, *Backward perturbation analysis of certain characteristic subspaces*, Numer. Math., 65 (1993), pp. 357–382.

[22] J. Weaver, *Centrosymmetric (cross-symmetric) matrices, their basic properties, eigenvalues, and eigenvectors*, Amer. Math. Monthly, 92 (1985), pp. 711–717.

[23] W. Wonham, *Linear Multivariable Control: A Geometric Approach*, Springer-Verlag, New York, 1979.

[24] D. Xie, X. Hu, and L. Zhang, *The solvability conditions for inverse eigenproblem of symmetric and anti-persymmetric matrices and its approximation*, Numer. Linear Algebra Appl., 10 (2003), pp. 223–234.

[25] S. Xu, *An Introduction to Inverse Algebraic Eigenvalue Problems*, Peking University Press, Peking, China, Vieweg and Sohn Publishing, Braunschweig, Germany, 1998.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 4, pp. 1115–1131

NECESSARY AND SUFFICIENT CONDITIONS FOR THE EXISTENCE OF POSITIVE DEFINITE SOLUTIONS TO THE SYMMETRIC RECURSIVE INVERSE EIGENVALUE PROBLEM*

ZHENYUE ZHANG†, JING WANG†, AND MIN FANG†

Abstract. Necessary and sufficient conditions are completely characterized for the existence of a positive definite or positive semidefinite solution to the symmetric recursive inverse eigenvalue problem (SRIEP). When a prior (indefinite) solution A to the SRIEP is known, positive definite/semidefinite solutions are formulated in terms of A and basis matrices of the column space of the given recursive matrix R and the null space of R^T. Taking into account some computational concerns, an algorithm is proposed that can check whether the SRIEP has a positive definite/semidefinite solution and find such a solution if it exists. Several numerical experiments are given to illustrate the performance of the algorithm.

Key words. inverse eigenvalue problem, positive definite/semidefinite solution, algorithm

AMS subject classifications. 15A29, 15A18, 15A49, 15A57

DOI. 10.1137/S0895479803431338

1. Introduction.

Given n real scalars $s_1, s_2, \ldots, s_n$ and n column vectors $r_k \in \mathcal{R}^k$, $k = 1, 2, \ldots, n$, the real symmetric recursive inverse eigenvalue problem (SRIEP) (of order n) is to find a symmetric matrix $A \in \mathcal{R}^{n\times n}$ such that

$$A_k r_k = s_k r_k, \quad k = 1, 2, \ldots, n, \tag{1.1}$$

where A_k is the kth leading principle submatrix of A. The SRIEP is one of the several recursive inverse eigenvalue problems discussed in [1]. In the special case when the upper triangular matrix R consisting of the vector sequence $\{r_k\}$,

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix} \quad \text{with} \quad r_k = \begin{bmatrix} r_{1k} \\ r_{2k} \\ \vdots \\ r_{kk} \end{bmatrix}, \quad k = 1, 2, \ldots, n,$$

is nonsingular, it is known that the SRIEP has a unique solution [1]. The unique solution is formulated as $A = R^{-T}(S \circ (R^T R))R^{-1}$, where S is characterized by the given eigenvalues $\{s_j\}$ as the matrix

$$S = \begin{bmatrix} s_1 & s_2 & s_3 & \cdots & s_n \\ s_2 & s_2 & s_3 & \cdots & s_n \\ s_3 & s_3 & s_3 & \cdots & s_n \\ \vdots & & & & \vdots \\ s_n & s_n & s_n & \cdots & s_n \end{bmatrix}$$

*Received by the editors July 7, 2003; accepted for publication (in revised form) by H. J. Woerdeman August 27, 2004; published electronically May 6, 2005.
http://www.siam.org/journals/simax/26-4/43133.html

†Department of Mathematics, Zhejiang University, Yuquan Campus, Hangzhou 310027, People's Republic of China (zyzhang@zju.edu.cn). The work of the first author was supported in part by the Special Funds for Major State Basic Research Projects of China (project G19990328) and NSFC project 60372033.

and $\circ$ denotes the Hadamard (or elementwise) product of matrices. It was proven in [1] that the SRIEP of order n has a solution if and only if the SRIEP of order $n-1$ has a solution, say A_{n-1}, such that the linear system

$$r_{nn}\alpha = s_n\tilde{r} - A_{n-1}\tilde{r}, \quad \tilde{r}^T\alpha + r_{nn}\beta = s_n r_{nn}$$

has a solution (α, β) with $\alpha \in \mathcal{R}^{n-1}$ and $\beta \in \mathcal{R}$, where $\tilde{r} = [r_{1,n}, \ldots, r_{n-1,n}]^T$. When R is singular, the *recursive* conditions above are too difficult to be verified in practice. Indeed, if R is singular and the SRIEP is solvable, then there are always multiple solutions.

What we are interested in is the existence question of a *positive definite* or *semidefinite* solution to the SRIEP. It is known that when R is nonsingular, the unique solution A is positive definite (semidefinite) if and only if $S \circ (R^T R)$ is positive definite (semidefinite). When R is singular, however, the existence question has not been answered completely. In [1], a sufficient condition for positive definite (semidefinite) solutions was proposed, stating that all r_k's are nonzero and $\{s_k\}$ satisfies the decreasing order

$$s_1 > s_2 > \cdots > s_{n-1} > s_n$$

and $s_n > 0$ ($s_n \geq 0$). In fact, this condition implies a stronger result that all solutions of the SRIEP are positive definite (semidefinite). It was conjectured in [1] that a positive semidefinite solution exists if and only if the SRIEP has a solution and $S \circ (R^T R)$ is positive semidefinite. This conjecture was shown later in [4], using counterexamples, to be incorrect generally. This conjecture would be true under the rank condition

$$\operatorname{rank}(S \circ (R^T R)) \geq n - 1, \tag{1.2}$$

implying that R has the null space of dimension at most one (see [4] for a proof).

One of our contributions in this paper is to show that the SRIEP may have positive definite or semidefinite solutions even if the rank condition (1.2) is not satisfied, provided that the SRIEP is solvable. Indeed, under the assumption that the SRIEP is solvable and $S \circ (R^T R)$ is positive semidefinite, we will prove the much stronger result that the SRIEP has a positive definite solution if and only if $\operatorname{rank}(S \circ (R^T R)) = \operatorname{rank}(R)$. Necessary and sufficient conditions for existence of a positive semidefinite solution are also given in this paper.

Though it is not our intention to touch upon the existence question of indefinite solutions in this paper, a simple structure for arbitrary (definite or indefinite) solutions will be presented. By exploiting this structure, one can construct a class of positive definite solutions to the SRIEP whenever the necessary and sufficient condition $\operatorname{rank}(S \circ (R^T R)) = \operatorname{rank}(R)$ is satisfied. In contrast to the positive definite case, it is difficult to construct a positive semidefinite solution to the SRIEP if a rank constraint to be described later does not hold—an overdetermined linear system will be involved in order to find a positive semidefinite solution. In short, our formulae for positive definite/semidefinite solutions are characterized in terms of an arbitrarily given solution to the SRIEP and orthogonal basis matrices of the column space of R and of the null space of R^T, respectively, and a solution of the overdetermined linear system if necessary.

Some computational issues will be considered for the purpose of simplifying the formulae and reducing the computational cost. We propose an algorithm that can

check whether the SRIEP has a positive definite or semidefinite solution. This algorithm can transform a given (indefinite) solution to a positive definite solution or a positive semidefinite solution (if it exists). Several numerical experiments will be given to illustrate the performance of the proposed algorithm.

Notation. In the rest of this paper, we will use $Q_1 \in \mathcal{R}^{n\times m}$ to denote henceforth an orthogonal basis matrix for the column space of R and write $R = Q_1 G_1^T$ with $G_1 \in \mathcal{R}^{n\times m}$ being of full column rank. This full-rank decomposition can be obtained via the QR decomposition with column pivoting [2, 3]. Furthermore, Q_1 can be constructed by means of the singular value decomposition (SVD) [3] where $Q_1^T A Q_1 = \mathrm{diag}(0, T_{11})$ with nonsingular T_{11}; see section 4 for the implementation of such a decomposition. Likewise, we shall use $Q_2 \in \mathcal{R}^{n\times(n-m)}$ to denote an orthogonal basis matrix for the null space of R^T when $m = \mathrm{rank}(R) < n$. Finally, we will denote by $Q = [Q_1, Q_2]$ an orthogonal matrix of order n.

2. Structure of a solution to the SRIEP. The following simple proposition given in [1] will be used often in our analysis for the existence of a solution to the SRIEP.

PROPOSITION 2.1. *If A is a solution to the SRIEP, then*

$$R^T A R = S \circ (R^T R). \tag{2.1}$$

Clearly, if R is invertible, the necessary condition (2.1) is also sufficient for A to be a solution. When R is singular, a matrix satisfying (2.1) may not solve the SRIEP. Such an example with a zero r_k was given in [1]. It is not difficult to construct a nonzero vector sequence $\{r_k\}$ and a scalar sequence $\{s_k\}$ for which there is an A that satisfies (2.1) but does not solve the corresponding SRIEP.

For any two solutions A and $\tilde{A}$ of the SRIEP, the difference $Z = \tilde{A} - A$ satisfies $R^T Z R = 0$ obviously. The following lemma shows a simple but important structure of Z.

LEMMA 2.2. *Assume that A is a solution to the SRIEP. Then $\tilde{A} = A + Z$ also solves the SRIEP if and only if Z is symmetric and ZR is a strictly lower triangular matrix.*

Proof. Because $A_k r_k = s_k r_k$, $k = 1, 2, \ldots, n$, it is easy to verify that

$$[I_k, 0](ZR)e_k = Z_k r_k = \big((A+Z)_k - A_k\big) r_k = (A+Z)_k r_k - s_k r_k,$$

where Z_k is the kth leading principle submatrix of Z in the same manner as we have denoted for A_k. Therefore $A + Z$ is a solution to the SRIEP if and only if

$$[I_k, 0](ZR)e_k = 0, \quad k = 1, 2, \ldots, n;$$

i.e., ZR is a strictly lower triangular matrix. ∎

Remark. The condition $R^T Z R = 0$ is automatically satisfied when Z is symmetric and ZR is strictly lower triangular because $R^T Z R$ is both strictly lower triangular and symmetric.

A symmetric matrix Z can be characterized by its entries in the upper triangular part. We denote by z the column vector consisting of these entries, and

$$z = [z_{11}, z_{12}, \ldots, z_{1n}, z_{22}, \ldots, z_{nn}]^T = [z_1, z_2, \ldots, z_n]^T,$$

where $z_k = [z_{k,k}, z_{k,k+1}, \ldots, z_{k,n}]$ is the kth row vector in the upper triangular part of Z. As we will show later, there is a lower triangular matrix C of order $n(n+1)/2$

such that Z is symmetric and ZR is strictly lower triangular if and only if the vector z solves the linear system

$$Cz = 0. \tag{2.2}$$

In fact, ZR is strictly lower triangular if and only if

$$e_k^T ZR[e_k, \dots, e_n] = [z_{k,1}, \dots, z_{k,k-1}, z_k] \begin{bmatrix} r_{1,k:n} \\ \vdots \\ r_{k-1,k:n} \\ R^{(k)} \end{bmatrix} = 0, \quad k = 1, \dots, n, \tag{2.3}$$

where $r_{i,k:n} = [r_{i,k}, r_{i,k+1}, \dots, r_{i,n}]$ and $R^{(k)} = (r_{i,j})_{i,j=k,\dots,n}$ is a submatrix of R.

To give the system (2.2), let us denote by $e_{k-i+1}^{(n-i+1)}$ the $(k-i+1)$th column of the unit matrix I_{n-i+1} of order $n-i+1$ and write $z_{ki} = z_{ik} = z_i e_{k-i+1}^{(n-i+1)}$ for $i < k$. We express the kth row $[z_{k,1}, \dots, z_{k,k-1}, z_k]$ of Z as

$$[z_1 e_k^{(n)}, z_2 e_{k-1}^{(n-1)}, \dots, z_{k-1} e_2^{(n-k-2)}, z_k] = z^T \begin{bmatrix} \operatorname{diag}(e_k^{(n)}, \dots, e_2^{(n-k-2)}, I_{n-k+1}) \\ 0 \end{bmatrix}.$$

Substituting the representation above into (2.3) gives $z^T c_k = 0$ with

$$c_k = \begin{bmatrix} e_k^{(n)} r_{1,k:n} \\ \vdots \\ e_2^{(n-k+2)} r_{k-1,k:n} \\ R^{(k)} \\ 0 \end{bmatrix} = 0, \quad k = 1, \dots, n.$$

Thus the coefficient matrix C in (2.2) is characterized by vectors c_k, $k = 1, \dots, n$,

$$C = [c_1, c_2, \dots, c_n]^T = \begin{bmatrix} R^{(1)} & e_2^{(n)} r_{1,2:n} & e_3^{(n)} r_{1,3:n} & \dots & e_n^{(n)} r_{1,n} \\ & R^{(2)} & e_2^{(n-1)} r_{2,3:n} & \dots & e_{n-1}^{(n-1)} r_{2,n} \\ & & R^{(3)} & \dots & e_{n-2}^{(n-2)} r_{3,n} \\ & & & \ddots & \vdots \\ & & & & R^{(n)} \end{bmatrix}^T.$$

Clearly, the upper triangular matrix C is sparse: each upper off-diagonal block is zero except one row.

The linear system (2.2) will be used to construct an (indefinite) solution for numerical testing. However, we do not suggest solving (2.2) directly for Z such that $A + Z$ is positive definite/semidefinite, taking the following into consideration:

(1) There is a simpler construction shown in (2.4) for Z that makes $A + Z$ a solution to the SRIEP. Indeed, under some conditions, one can construct a W in (2.4) that ensures $A + Z$ is positive definite or semidefinite.

(2) To ensure that $A + Z$ with Z retrieved by a special solution z to (2.2) is positive definite or semidefinite, some necessary constraints should be imposed to (2.2). However, it is very difficult to impose such constraints. This obstruction can be avoided by implicitly imposing a rank constraint to (2.2); i.e., we just look for a solution z to (2.2) such that $\hat{A} = A + Z$ satisfies a rank condition. With $\hat{A}$, one can construct a positive semidefinite solution easily. We will show the details in section 5.

LEMMA 2.3. *Let $m = \text{rank}(R)$ and let A be a given solution of the SRIEP. Corresponding to any symmetric matrix $W \in \mathcal{R}^{(n-m)\times(n-m)}$, define*

$$Z = Q_2 W Q_2^T \tag{2.4}$$

with an orthogonal basis matrix Q_2 of the null space R^T. Then $A + Z$ is a solution to the SRIEP.

Proof. The result follows immediately by Lemma 2.2 because $ZR = 0$. ∎

Though the set of all symmetric matrices in the form $A + Q_2 W Q_2^T$ may not cover all the solutions of the SRIEP, we can prove that this subset contains a positive definite matrix if the SRIEP has a positive definite solution; that is, one can choose a symmetric W such that $A + Q_2 W Q_2^T$ is positive definite. In the next section, we will show how to construct such a matrix W as required. Furthermore, in the case when only positive semidefinite solutions exist, such a solution $\tilde{A} = A + Z$ can also be found with Z in the same form $Z = Q_2 W Q_2^T$, provided that a certain rank condition relative to A holds. However, when this rank condition is not satisfied, it will be difficult to construct a symmetric Z such that $A + Z$ is a positive semidefinite solution to the SRIEP. In section 4 we will give a detailed analysis of constructing positive semidefinite solutions.

3. Necessary and sufficient conditions for positive definite solutions. Proposition 2.1 implies the following necessary conditions for the existence of positive definite or semidefinite solutions:

$$S \circ (R^T R) \text{ is positive semidefinite.} \tag{3.1}$$

$$\text{rank}(S \circ (R^T R)) \le \text{rank}(R). \tag{3.2}$$

In fact, the equality must hold in (3.2) if a positive definite solution exists, as we will see in the proof below.

THEOREM 3.1. *Assume that the SRIEP has a solution and that $S \circ (R^T R)$ is positive semidefinite. Then the SRIEP has a positive definite solution if and only if*

$$\text{rank}(S \circ (R^T R)) = \text{rank}(R). \tag{3.3}$$

Furthermore, if (3.3) *holds and A is an arbitrary solution to the SRIEP, then for any positive definite matrix W_0,*

$$\tilde{A} = Q_2 W_0 Q_2^T + A Q_1 (Q_1^T A Q_1)^{-1} Q_1^T A \tag{3.4}$$

is a positive definite solution to the SRIEP.

Proof. We first prove the sufficiency. Let $m = \text{rank}(R) < n$. By Lemma 2.3, if A is a solution to the SRIEP, then for any symmetric $W \in \mathcal{R}^{(n-m)\times(n-m)}$, $A + Q_2 W Q_2^T$ is also a solution. What we need to show is that a symmetric W exists such that $A + Q_2 W Q_2^T$ is positive definite. To this end, let $R = Q_1 G_1^T$ be a full-rank decomposition of R with $G_1 \in \mathcal{R}^{n\times m}$ having full column rank. Denoting by $Q = [Q_1, Q_2]$, the orthogonal matrix gives

$$Q^T A Q = \begin{bmatrix} Q_1^T A Q_1 & Q_1^T A Q_2 \\ Q_2^T A Q_1 & Q_2^T A Q_2 \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix},$$

where $T_{ij} = Q_i^T A Q_j$. We have

$$\begin{aligned} Q^T (A + Q_2 W Q_2^T) Q &= \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} + W \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ T_{21} T_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} T_{11} & 0 \\ 0 & W + T_{22} - T_{21} T_{11}^{-1} T_{12} \end{bmatrix} \begin{bmatrix} I & T_{11}^{-1} T_{12} \\ 0 & I \end{bmatrix}. \end{aligned} \tag{3.5}$$

Note that $S \circ (R^T R) = R^T A R = G_1 T_{11} G_1^T$. It follows that T_{11} is at least positive semidefinite because $S \circ (R^T R)$ is positive semidefinite. Indeed, the rank condition (3.3) implies that T_{11} is nonsingular. Hence T_{11} is positive definite. By (3.5), it follows that $A + Q_2 W Q_2^T$ is positive definite if and only if $W + T_{22} - T_{21} T_{11}^{-1} T_{12}$ is positive definite. For any positive definite matrix W_0, one can set $W = W_0 - T_{22} + T_{21} T_{11}^{-1} T_{12}$ as required. We can simplify the representation $\tilde{A} = A + Q_2 W Q_2^T = A + Q_2 (W_0 - T_{22} + T_{21} T_{11}^{-1} T_{12}) Q_2^T$ to (3.4). In fact, substituting $T_{ij} = Q_i^T A Q_j$ into the representation above, we have that

$$\begin{aligned}
\tilde{A} &= Q_2 W_0 Q_2^T + A - Q_2 Q_2^T (A - A Q_1 T_{11}^{-1} Q_1^T A) Q_2 Q_2^T \\
&= Q_2 W_0 Q_2^T + A - (I - Q_1 Q_1^T)(A - A Q_1 T_{11}^{-1} Q_1^T A)(I - Q_1 Q_1^T) \\
&= Q_2 W_0 Q_2^T + A - (A - A Q_1 T_{11}^{-1} Q_1^T A) \\
&= Q_2 W_0 Q_2^T + A Q_1 T_{11}^{-1} Q_1^T A.
\end{aligned}$$

We next prove the necessity. Let A be a positive definite solution to the SRIEP. Then $Q_1^T A Q_1$ is nonsingular and

$$\operatorname{rank}(S \circ (R^T R)) = \operatorname{rank}(R^T A R) = \operatorname{rank}(Q_1^T A Q_1) = m = \operatorname{rank}(R).$$

The proof is now complete. ∎

As a corollary of Theorem 3.1, the main result of [4] stated below can be easily proved.

COROLLARY 3.2. *If the SRIEP is solvable and $S \circ (R^T R)$ is positive semidefinite with* $\operatorname{rank}(S \circ (R^T R)) \geq n - 1$, *then SRIEP has a positive semidefinite solution.*

Proof. We only need to prove the corollary in the case when R is singular. By Proposition 2.1, the assumption that SRIEP is solvable implies that

$$n - 1 \leq \operatorname{rank}(S \circ (R^T R)) \leq \operatorname{rank}(R) \leq n - 1,$$

implying that $\operatorname{rank}(S \circ (R^T R)) = \operatorname{rank}(R) = n - 1$. By Theorem 3.1, the SRIEP has a positive definite solution. ∎

Obviously, positive definite solutions are not unique if R is singular because of the free choice of W_0 in (3.4). Note that the positive solutions given in Theorem 3.1 have the form $\tilde{A} = A + Q_2 W Q_2^T$.

Remark. It is worth mentioning that the set of positive definite solutions given in Theorem 3.1 may not cover all the positive definite solutions; it is possible that for two solutions A and $\tilde{A}$ of the SRIEP the gap $Z = \tilde{A} - A$ may be different in form from $Q_2 W Q_2^T$, even if $\tilde{A}$ is positive definite. Below is such an example.

Example 1. Let $s_1 = 1$, $s_2 = 2$, $s_3 = 1$, and

$$r_1 = [1], \quad r_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad r_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 0 & 0 \\ & 0 & 0 \\ & & 1 \end{bmatrix}.$$

It is easy to verify that the SRIEP has a positive definite solution I and an indefinite solution

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

Clearly $Z = I - A$ does not have the form (2.4) or (3.4) because $ZR \neq 0$.

It is clear that if we choose a positive semidefinite W_0, the solution $\tilde{A}$ given in (3.4) is positive semidefinite, too, because the two matrices $\tilde{A} = A + Q_2(W_0 - T_{22} + T_{21}T_{11}^{-1}T_{12})Q_2^T$ and $\mathrm{diag}(T_{11}, W_0)$ have the same inertia, and the latter is obviously positive semidefinite. This means that positive semidefinite solutions to the SRIEP exist if the conditions of Theorem 3.1 hold. In the next section, we will give weaker necessary and/or sufficient conditions for the existence of a positive semidefinite solution to the SRIEP.

4. Necessary and sufficient conditions for positive semidefinite solutions. We first show a weaker rank-condition (see (4.4)) with which a positive semidefinite solution can be easily constructed as in the last section. The more difficult case when (4.4) is not satisfied will be considered later.

Denote $m_1 = \mathrm{rank}(R^TAR)$ and $m = \mathrm{rank}(R)$ as before. Because $S \circ R^TR = R^TAR$, the value m_1 is less than or equal to m and is invariant for all choices of A in the solution set of the SRIEP. Clearly if $m_1 = m$ for a solution A of the SRIEP, then the condition (3.3) holds and the existence question has been completely answered in the last section. If $m_1 < m$, we further assume that the orthogonal basis matrix Q_1 of the range space of R can be partitioned as $Q_1 = [Q_{01}, Q_{11}]$ with $Q_{01} \in \mathcal{R}^{n\times(m-m_1)}$ and $Q_{11} \in \mathcal{R}^{n\times m_1}$, such that $Q_1^TAQ_{01} = 0$ and $T_{11} = Q_{11}^TAQ_{11}$ is nonsingular. This partition gives

$$Q_1^TAQ_1 = \begin{bmatrix} 0 & \\ & Q_{11}^TAQ_{11} \end{bmatrix} = \begin{bmatrix} 0 & \\ & T_{11} \end{bmatrix}. \tag{4.1}$$

In fact, by the SVD, such an orthogonal basis matrix Q_1 can be easily obtained and T_{11} can be diagonal via the following steps:

(1) Compute an orthonormal matrix $\tilde{Q}_1$ by column pivoting QR to R [3].
(2) Compute the eigendecomposition of $\tilde{Q}_1^TA\tilde{Q}_1$, $\tilde{Q}_1^TA\tilde{Q}_1 = H_1\,\mathrm{diag}(0, \Lambda_1)H_1^T$, where Λ_1 is nonsingular and diagonal, H_1 is orthogonal.
(3) Set $Q_1 = \tilde{Q}_1H_1$ and $T_{11} = \Lambda_1$.

In the rest of this paper, we always assume that (4.1) holds. We remark that (4.1) is always true for any other solution, say $\hat{A}$ to the SRIEP, because $R^T\hat{A}R = S\circ(R^TR) = R^T\hat{A}R$.

To facilitate our analysis, we need the following lemma.

LEMMA 4.1. *If $m_1 < m$ and (4.1) holds with $Q_1 = [Q_{01}, Q_{11}]$, then*

$$\mathrm{rank}(AR) = \mathrm{rank}(R^TAR) + \mathrm{rank}(Q_2^TAQ_{01}). \tag{4.2}$$

Proof. Obviously by (4.1),

$$Q^TAQ_1 = \begin{bmatrix} Q_1^TAQ_1 \\ Q_2^TAQ_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & Q_{11}^TAQ_{11} \\ Q_2^TAQ_{01} & Q_2^TAQ_{11} \end{bmatrix}. \tag{4.3}$$

Because $Q_{11}^TAQ_{11}$ is nonsingular, we have that

$$\mathrm{rank}(AR) = \mathrm{rank}(Q^TAQ_1) = \mathrm{rank}(Q_{11}^TAQ_{11}) + \mathrm{rank}(Q_2^TAQ_{01}).$$

Then (4.2) follows immediately since $\mathrm{rank}(R^TAR) = \mathrm{rank}(Q_{11}^TAQ_{11})$. ∎

Remark. Because for any two solutions A and $\tilde{A}$ to the SRIEP, it is always true that $R^TAR = R^T\tilde{A}R$, it follows that (4.1) holds for $\tilde{A}$ if and only if it holds for A.

THEOREM 4.2. *Assume that $S \circ (R^TR)$ is positive semidefinite.*

(1) *If the SRIEP has an (indefinite) solution A, and*

$$\operatorname{rank}(R^T AR) = \operatorname{rank}(AR), \tag{4.4}$$

then for any positive definite (or semidefinite) matrix W_0,

$$\tilde{A} = Q_2 W_0 Q_2^T + AQ_{11}(Q_{11}^T AQ_{11})^{-1} Q_{11}^T A \tag{4.5}$$

is also a positive semidefinite solution to the SRIEP.

(2) *The equality* (4.4) *holds for any positive semidefinite solution A of the SRIEP.*

Proof. By Lemma 4.1, condition (4.4) implies $Q_2^T AQ_{01} = 0$. Because $Q_1^T AQ_{01} = 0$, we have $AQ_{01} = 0$. It follows that

$$Q^T AQ = \begin{bmatrix} 0 & & \\ & Q_{11}^T AQ_{11} & Q_{11}^T AQ_2 \\ & Q_2^T AQ_{11} & Q_2^T AQ_2 \end{bmatrix} \equiv \begin{bmatrix} 0 & & \\ & T_{11} & T_{12} \\ & T_{21} & T_{22} \end{bmatrix}. \tag{4.6}$$

Recall that $T_{11} = Q_{11}^T AQ_{11}$ is nonsingular and

$$S \circ (R^T R) = R^T AR = G_1 Q_1^T AQ_1 G_1^T = G_1 \operatorname{diag}(0, T_{11}) G_1^T$$

is positive semidefinite. We conclude that T_{11} is positive definite. By the proof of Theorem 3.1,

$$B = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} + W \end{bmatrix}$$

is at least positive semidefinite if $W = W_0 - T_{22} + T_{21} T_{11}^{-1} T_{12}$ with any positive definite or semidefinite $W_0 \in \mathcal{R}^{(n-m)\times(n-m)}$. Therefore $\tilde{A} = A + Q_2 WQ_2^T = Q \operatorname{diag}(0,\ B) Q^T$ is positive semidefinite, too. Recalling that $AQ_1 = [0, AQ_{11}]$, similarly we have

$$\begin{aligned} \tilde{A} &= A + Q_2 WQ_2^T \\ &= A + Q_2 W_0 Q_2^T - (I - Q_1 Q_1^T)(A - AQ_{11} T_{11}^{-1} Q_{11}^T A)(I - Q_1 Q_1^T) \\ &= Q_2 W_0 Q_2^T + AQ_{11} T_{11}^{-1} Q_{11}^T A, \end{aligned}$$

yielding (4.5).

Furthermore, consider

$$[Q_{01},\ Q_2]^T A [Q_{01},\ Q_2] = \begin{bmatrix} 0 & Q_{01}^T AQ_2 \\ Q_2^T AQ_{01} & Q_2^T AQ_2 \end{bmatrix}.$$

Because A is positive semidefinite, we have $Q_2^T AQ_{01} = 0$. Thus (4.4) is true by Lemma 4.1. ∎

The following example illustrates Theorems 3.1 and 4.2.

Example 2. Let $n = 3$, $s_1 = s_2 = 1$, $s_3 = 1 + s$, and

$$r_1 = [1], \quad r_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad r_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 1 & 1 \\ & 0 & 1 \\ & & 1 \end{bmatrix}.$$

It is easy to verify that for any given s, A is a solution to the SRIEP if and only if there is a real t such that

$$A = \begin{bmatrix} 1 & 0 & s \\ 0 & s+t & 1-t \\ s & 1-t & t \end{bmatrix}.$$

Simple calculations give that

$$AR = \begin{bmatrix} 1 & 1 & 1+s \\ 0 & 0 & 1+s \\ s & s & 1+s \end{bmatrix}, \quad R^T AR = \begin{bmatrix} 1 & 1 & 1+s \\ 1 & 1 & 1+s \\ 1+s & 1+s & 3(1+s) \end{bmatrix}.$$

(1) If $(1+s)(2-s) > 0$, i.e., if $-1 < s < 2$, then $R^T AR$ is positive semidefinite and

$$\operatorname{rank}(R^T AR) = \operatorname{rank}(AR) = \operatorname{rank}(R) = 2.$$

It is not difficult to verify that $\det(A) = \big(t(2-s) - (1-s+s^2)\big)(1+s)$. Denote $\eta = (1-s+s^2)/(2-s) > 0$. It follows that A is positive definite for $t > \eta$, positive semidefinite for $t = \eta$, or indefinite if $t < \eta$. It verifies Theorems 3.1 and 4.2.

(2) If $(1+s)(2-s) < 0$, then $R^T AR = S \circ (R^T R)$ is indefinite and the SRIEP has no positive definite/semidefinite solutions, even though $\operatorname{rank}(R^T AR) = \operatorname{rank}(AR) = \operatorname{rank}(R)$. This example also shows that these rank equalities do not imply the positive semidefiniteness of $S \circ (R^T R)$.

(3) If $s = -1$, then $1 = \operatorname{rank}(R^T AR) = \operatorname{rank}(AR) < \operatorname{rank}(R) = 2$. Because $\det(A) = 0$ for all t, the SRIEP does not have a positive definite solution, as is shown by Theorem 3.1. However, positive semidefinite solutions ($t \geq 1$) and indefinite solutions ($t < 1$) exist.

(4) Finally, if $s = 2$, $R^T AR = S \circ (R^T R)$ is also positive semidefinite, and

$$1 = \operatorname{rank}(R^T AR) < \operatorname{rank}(AR) = \operatorname{rank}(R) = 2.$$

It is not difficult to verify that for any t, A is always indefinite because the first diagonal of A is positive and $\det(A) < 0$. So if condition (4.4) does not hold, SRIEP may have no positive semidefinite solutions.

For a given solution A to the SRIEP, the rank condition (4.4) is not necessary for the existence of positive semidefinite solutions. Below is evidence for the phenomena.

Example 3. Let $n = 3$, $s_1 = 1$, $s_2 = 0$, $s_3 = 2$, and

$$r_1 = [1], \quad r_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad r_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & -1 & 1 \\ & 1 & 1 \\ & & 0 \end{bmatrix}.$$

It is easy to verify that indefinite matrix

$$A = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix}$$

is a solution of the SRIEP but does not satisfy the rank constraint (4.4). However, the SRIEP has a positive semidefinite solution $A + Z$ with

$$Z = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix}.$$

We further notice that the modification matrix Z above cannot be written in the form $Q_2 W Q_2^T$ since $\operatorname{rank}(Z) = 2$ and $\operatorname{rank}(Q_2 W Q_2^T) \leq 1$ for all W.

However, if the rank constraint (4.4) does not hold, it will be more difficult to directly construct a Z such that $A+Z$ is a positive semidefinite solution to the SRIEP. In fact, for any W, $(A + Q_2WQ_2^T)R = AR$. If (4.4) does not hold for A, then it also does not hold for $A+Q_2WQ_2^T$. Thus by Theorem 4.2, for all choices of W, $A+Q_2WQ_2^T$ cannot be a positive semidefinite solution to the SRIEP, meaning that a more general form should be considered for Z if one wants to find a positive semidefinite solution in the form $A + Z$. We have mentioned in section 2 that the problem of finding such a Z is equivalent to the problem of solving the upper triangular system (2.2) for the vector z that consists of the entries in the upper triangular part of Z. We also pointed out that it is very difficult to impose constraints to the linear system to ensure the positive semidefiniteness of $A + Z$ with Z retrieved by a solution z. To deal with this problem, we will update A to $\hat{A} = A + Z_0$ with a symmetric Z_0 such that $\hat{A}$ is also a solution to the SRIEP and the rank condition (4.4) holds for $\hat{A}$. This solution $\hat{A}$ may also be indefinite. As we have shown, as soon as such an $\hat{A}$ is available, it is ready to construct a positive semidefinite solution $\hat{A} + Q_2WQ_2^T$. In the proof of the following theorem, we will show that such a symmetric Z_0 can be obtained by solving the linear system.

THEOREM 4.3. *Assume that $S\circ(R^TR)$ is positive semidefinite and A is a solution of the SRIEP. Then the SRIEP has a positive semidefinite solution if and only if there exists a strictly lower triangular matrix L that satisfies*

$$Q_1^TL = 0, \quad LP = -AQ_{01}Q_{01}^TRP, \tag{4.7}$$

where $P \in \mathcal{R}^{n\times(n-m_1)}$ is an orthogonal basis matrix of the null space of Q_{11}^TR.

Furthermore, if the strictly lower triangular matrix L satisfies (4.7)*, then corresponding to any positive semidefinite W_0, a positive semidefinite solution to the SRIEP is given by*

$$\tilde{A} = Q_2W_0Q_2^T + (AQ_{11} + E)(Q_{11}^TAQ_{11})^{-1}(AQ_{11} + E)^T, \tag{4.8}$$

where $E = (L + AQ_{01}Q_{01}^TR)(R^TQ_{11})^\dagger$ and $\dagger$ denotes the Moore–Penrose generalized inverse of a matrix.

Proof. Assume that the SRIEP has a positive semidefinite solution $\tilde{A}$. Lemma 2.2 shows that $L = (\tilde{A} - A)R$ is a strictly lower triangular matrix. We now prove that L satisfies (4.7).

First, by Proposition 2.1, the first equality in (4.7) follows from $R^TL = R^T(\tilde{A} - A)R = 0$ immediately. Second, (2.4) remains true when A is replaced by $\tilde{A}$ because $R^T\tilde{A}R = R^TAR$. By Lemma 4.1, $\operatorname{rank}(\tilde{A}R) = \operatorname{rank}(R^T\tilde{A}R)+\operatorname{rank}(Q_2^T\tilde{A}Q_{01})$. On the other hand, since $\tilde{A}$ is a positive semidefinite solution, we have by Theorem 4.2 that $\operatorname{rank}(R^T\tilde{A}R) = \operatorname{rank}(\tilde{A}R)$. Thus $Q_2^T\tilde{A}Q_{01} = 0$, implying that $\tilde{A}Q_{01} = 0$. Therefore the difference $Z = \tilde{A} - A$ satisfies that $ZQ_{01} = -AQ_{01}$ and

$$L = ZR = Z[Q_{01}, Q_{11}]Q_1^TR = -AQ_{01}Q_{01}^TR + ZQ_{11}Q_{11}^TR.$$

It follows that for an orthogonal basis matrix P of the null space of Q_{11}^TR, the second equality $LP = -AQ_{01}Q_{01}^TRP$ in (4.7) holds because $(Q_{11}^TR)P = 0$. This proves the necessity.

We now prove the sufficiency. Let L be a strictly lower triangular matrix satisfying (4.7). Denote $G_{11} = R^TQ_{11}$ and

$$W = (L + AQ_{01}Q_{01}^TR)G_{11}(G_{11}^TG_{11})^{-1}Q_{11}^T. \tag{4.9}$$

It is easy to verify $Q_1^T W = 0$ because $Q_1^T L = 0$ and $Q_1^T A Q_{01} = 0$. Thus the symmetric matrix Z defined by

$$Z = -AQ_{01}Q_{01}^T - Q_{01}Q_{01}^T A + W + W^T \tag{4.10}$$

satisfies $ZR = -AQ_{01}Q_{01}^T R + WR$. On the other hand, because P is an orthogonal basis matrix of the null space of G_{11}^T, we have $G_{11}(G_{11}^T G_{11})^{-1} G_{11}^T = I - PP^T$. It follows from (4.9) and (4.7) that

$$WR = (L + AQ_{01}Q_{01}^T R)(I - PP^T) = L + AQ_{01}G_{01}^T.$$

Therefore, $ZR = -AQ_{01}Q_{01}^T R + WR = L$ is strictly lower triangular. By Lemma 2.2,

$$\hat{A} = A + Z = A - AQ_{01}Q_{01}^T - Q_{01}Q_{01}^T A + W + W^T$$

solves the SRIEP.

Now we show that $\operatorname{rank}(R^T \hat{A} R) = \operatorname{rank}(\hat{A}R)$ holds. In fact, by the definition (4.9) of W, $WQ_{01} = 0$ and $W^T Q_{01} = 0$. Thus by (2.4), $ZQ_{01} = -AQ_{01}$, giving $\hat{A}Q_{01} = 0$. Therefore by Lemma 4.1, the equality $\operatorname{rank}(R^T \hat{A} R) = \operatorname{rank}(\hat{A}R)$ is true.

Finally, applying Theorem 4.2 to $\hat{A}$ and using $Q_{11}^T \hat{A} Q_{11} = Q_{11}^T A Q_{11}$, we have the positive semidefinite solution

$$\tilde{A} = Q_2 W_0 Q_2^T + \hat{A}Q_{11}(Q_{11}^T A Q_{11})^{-1} Q_{11}^T \hat{A} \tag{4.11}$$

to the SRIEP for any positive definite matrix W_0. Here we have used $Q_{11}^T \hat{A} Q_{11} = Q_{11}^T A Q_{11}$. The formula (4.11) of $\tilde{A}$ can be simplified. In fact, by $W^T Q_1 = 0$,

$$\hat{A}Q_{11} = (A + W)Q_{11} = AQ_{11} + (L + AQ_{01}Q_{01}^T R)G_{11}(G_{11}^T G_{11})^{-1} = AQ_{11} + E.$$

Substituting $\hat{A}Q_{11} = AQ_{11} + E$ into (4.11) yields (4.8) immediately. □

In the next section, we will discuss some numerical computational issues for solving the linear system (4.7) and propose our algorithm for computing a positive definite/semidefinite solution to the SRIEP.

5. Computational issues and an algorithm. The formulae for positive definite/semidefinite solutions to the SRIEP proposed in Theorems 3.1 and 4.2 and Lemma 4.1 are very simple and easy to implement. However, the formula (4.8) contains a solution to the linear system (4.7). In this section, we will discuss how to reduce the computational complexity of solving the linear system (4.7). A complete algorithm for the construction of positive definite/semidefinite solutions to the SRIEP will be posed later.

Obviously, (4.7) is equivalent to a linear system in vector form

$$Fx = -b \tag{5.1}$$

with vector x consisting of $n(n-1)/2$ entries in the strictly lower triangular part of L. Notice that $Q_1^T L = 0$ has $m(n-1)$ nontrivial equations, and the linear system $LP = -AQ_{01}Q_{01}^T RP$ contains $(n-1)(n-m_1)$ equations, ignoring the first row of the matrix equation. Such an equivalent system (5.1) should have $(n-1)(n+m-m_1)$ equations; i.e., F should be a matrix of order $(n-1)(n+m-m_1) \times n(n-1)/2$. We are not going to show the construction for F in detail because, taking into account the advantage of (4.7), the large scale of (5.1) can be reduced much and an equivalent smaller system will been given later in detail.

To this end, we partition L, Q_1, and P as follows:

$$L = \begin{bmatrix} 0 & 0 \\ X & 0 \end{bmatrix},\ Q_1 = \begin{bmatrix} q^T \\ U^T \end{bmatrix},\ P = \begin{bmatrix} H \\ h^T \end{bmatrix},$$

where q and h are column vectors. Also, denote by a^T the first row of A and partition $AQ_{01}Q_{01}^TRP$ as

$$AQ_{01}Q_{01}^TRP = \begin{bmatrix} a^TQ_{01}Q_{01}^TRP \\ B \end{bmatrix}$$

with $B = A(2:n,:)Q_{01}Q_{01}^TRP \in \mathcal{R}^{(n-1)\times(n-m_1)}$, where $A(2:n,:)$ denotes the submatrix of the second row through the last row in A. Then (4.7) is equivalent to

$$UX = 0, \quad XH = -B \quad \text{s.t. } X \in \mathcal{R}^{(n-1)\times(n-1)} \text{ is lower triangular} \tag{5.2}$$

together with

$$a^TQ_{01}Q_{01}^TRP = 0. \tag{5.3}$$

Clearly (5.3) gives rise to a necessary condition for the existence of a positive semidefinite solution to the SRIEP.

Let $x_j = (x_{j,j}, x_{j+1,j}, \ldots, x_{n-1,j})^T$ be the jth column vector of the low triangular part of $X = (x_{ij})$; i.e., the jth column of X reads $Xe_j = [0, \ldots, 0, x_j^T]^T$. Denote by u_j the jth column of U and set $U_j = [u_j, \ldots, u_{n-1}]$. Thus the jth column of $UX = 0$ reads $U_jx_j = 0$, meaning that x_j belongs to the null space of U_j. Denoting by V_j an orthogonal basis matrix of the null space of U_j, we can write $x_j = V_jy_j$ if $k_j = \dim(V_j) > 0$; otherwise $x_j = 0$. Therefore, $UX = 0$ if and only if

$$x_j = \begin{cases} 0 & \text{if} \quad k_j = 0, \\ V_jy_j & \text{if} \quad k_j > 0, \end{cases} \qquad j = 1, \ldots, n-1. \tag{5.4}$$

On the other hand, we will rewrite $XH = -B$ in terms of the vectors x_i's. To this end, let's denote by b_i the ith column of B and $H = (h_{ij}) \in \mathcal{R}^{(n-1)\times(n-m_1)}$. The ith column equation of $XH = -B$ can be written as

$$h_{1,i}x_1 + \begin{bmatrix} 0 \\ h_{2,i}x_2 \end{bmatrix} + \cdots + \begin{bmatrix} 0 \\ h_{n-1,i}x_{n-1} \end{bmatrix} = -b_i.$$

Substituting $x_i = V_iy_i$ into the left side of the equality above gives

$$h_{1,i}V_1y_1 + \begin{bmatrix} 0 \\ h_{2,i}V_2 \end{bmatrix} y_2 + \cdots + \begin{bmatrix} 0 \\ h_{n-1,i}V_{n-1} \end{bmatrix} y_{n-1} = -b_i$$

or, equivalently,

$$[\Phi_{i,1}, \ldots, \Phi_{i,n-1}]y = -b_i, \quad i = 1, \ldots, n-m_1, \tag{5.5}$$

where y is the longer column vector of dimension $K = \sum_{i=1}^{n-1} k_i$, linked by all the vectors $y_1, \ldots, y_{n-1}$ one by one,

$$y = [y_1^T, y_2^T, \ldots, y_{n-1}^T]^T,$$

and

$$\Phi_{ij} = \begin{bmatrix} 0 \\ h_{ji}V_j \end{bmatrix} \in \mathcal{R}^{(n-1)\times k_j}, \quad i = 1, \ldots, n-m_1,\ j = 1, \ldots, n-1.$$

Note that the quantities Φ_{ij} and y_j disappear if $k_j = 0$. Now we set $\Phi = \big(\Phi_{ij}\big) \in \mathcal{R}^{(n-1)(n-m_1)\times K}$ and $b = [b_1^T, b_2^T, \ldots, b_{n-1}^T]^T$. Thus the system $XH = -B$ or (5.5) can be equivalently rewritten as

$$\Phi y = -b. \tag{5.6}$$

We formulate the construction as the following result.

THEOREM 5.1. *Assume that* (5.3) *holds and* y *is a solution to the linear system of* (5.6). *Partition* $y^T = [y_1^T, y_2^T, \ldots, y_{n-1}^T]$ *with* $y_j \in \mathcal{R}^{k_j}$ *for* $k_j = \dim(V_j)$, *where* V_j *is an orthogonal basis matrix of the null space of* U_j, $j = 1, \ldots, n-1$. *Define* $x_j = (x_{j+1,j}, \ldots, x_{n,j})^T$ *as in* (5.4). *Then the lower triangular matrix*

$$L = \begin{bmatrix} 0 & & & \\ x_{21} & 0 & & \\ \vdots & \ddots & \ddots & \\ x_{n1} & \ldots & x_{n,n-1} & 0 \end{bmatrix} \tag{5.7}$$

is a solution to (4.7).

In general, $K = \sum_{j=1}^{n-1} k_j \ll n(n-1)/2$ and $(n-1)(n-m_1) \ll (n-1)(n+m-m_1)$. Therefore, comparing its size to that of F in (5.1), Φ in (5.6) is much smaller. We will illustrate the saving through our numerical experiments in the next section. When (5.3) holds, SRIEP has a positive semidefinite solution if and only if (5.6) is solvable. As soon as a solution y to (5.6) is obtained, we can reconstruct x_j by (5.4), i.e., set x_j a zero vector of dimension $n-j$ if $k_j = 0$, or $x_j = V_j y_j$ if $k_j > 0$, as well as a strictly lower triangular matrix L by appending x_j's column by column in the lower triangular part, i.e., set the jth column of L as $Le_j = \left[\begin{smallmatrix} 0 \\ x_j \end{smallmatrix}\right]$.

Now we are ready to present our algorithm that transforms an indefinite solution to a positive definite/semidefinite solution of the SRIEP (see Figure 1).

6. Numerical experiments. The formulae (3.4), (4.5), and (4.8) clearly show that when R is singular and a positive definite/semidefinite solution to the SRIEP exists, then the SRIEP must have multiple positive definite/semidefinite solutions, because one can choose different positive definite matrix W_0. In our numerical experiments, we simply set $W_0 = I$.

The required data sets in our numerical experiments are constructed artificially. We first construct a positive definite matrix B as follows (using the notation of MATLAB):

```
[Q,temp] = qr(full(sprand(n,n,0.8)));
lambda = 1+10*rand(n,1);
B = Q*diag(lambda)*Q';
```

To construct a data set $\{s_k, r_k\}$ such that the upper triangular matrix R constructed by r_k's has at least a diagonal element as small as possible in absolute value (ideally, we want R to be singular), and B is a positive semidefinite solution to the corresponding SRIEP, i.e., (s_k, r_k) satisfies $B_k r_k = s_k r_k$ for the kth principle submatrix B_k of B, $k = 1, \ldots, n$, we compute an eigenvalue decomposition $B_k =$

Algorithm. Given a solution A of the SRIEP with singular R, this algorithm transforms A to a positive definite/semidefinite solution if it exists.

1. Compute an orthogonal basis matrix $Q_1 = [Q_{01}, Q_{11}]$ of the column space of R such that $[Q_{01}, Q_{11}]^T A[Q_{01}, Q_{11}] = \mathrm{diag}(\Lambda_0, \Lambda_1)$ with zero or empty Λ_0 and nonsingular Λ_1.

2. Check the existence.
- **2.1** If Λ_1 has negative diagonal elements, the SRIEP does not have positive definite or positive semidefinite solutions, and terminate.
- **2.2** If Λ_0 exists, the SRIEP has no positive definite solutions.
- **2.3** If Λ_0 is empty and all diagonal elements of Λ_1 are positive, the SRIEP has positive definite solutions.

3. If Λ_0 is empty, compute $\tilde{A}$ by (3.4), and terminate.

4. If Λ_0 exists and $AQ_{01} = 0$, compute $\tilde{A}$ by (4.5), and terminate.

5. If Λ_0 exists and $AQ_{01} \neq 0$,
- **5.1** Check the necessary condition (5.3). If it does not hold, the SRIEP has no positive semidefinite solutions, and terminate.
- **5.2** Compute the orthogonal basis matrices V_j's of the null spaces of U_j's.
- **5.3** If (5.6) is not solvable, the SRIEP has no positive semidefinite solutions, and terminate,
- **5.4** Solve (5.6) to obtain y, retrieve x_i's by (5.4), and construct the strictly lower triangular L by (5.7).
- **5.5** Compute $\tilde{A}$ by (4.8).

FIG. 1. *Algorithm for computing positive definite/semidefinite solutions of the SRIEP.*

$U^{(k)}\,\mathrm{diag}(\lambda_1^{(k)}, \ldots, \lambda_1^{(k)})(U^{(k)})^T$ of B_k and select an eigenpair $(\lambda_j^{(k)}, u_j^{(k)})$ as (s_k, r_k) if $|u_{kj}^{(k)}| = \min_{1\le i\le k} |u_{ki}^{(k)}|$, where $u_{kj}^{(k)}$ is the last component of the kth unit column $u_j^{(k)}$ in $U^{(k)}$. To construct an indefinite solution $A = B + Z$ to the SRIEP with respect to the data set $\{s_k, r_k\}$, we select a symmetric Z such that ZR is strictly lower triangular (see the description given in section 2) and by verifying whether $A = B + Z$ has at least a negative eigenvalue. Those data sets $\{s_k, r_k\}$ and the indefinite A will be adopted if $\min_i(|r_{ii}|) < 10^{-13}$ and A has an eigenvalue less than $-0.1\|A\|_2$. Then we apply our algorithm to the data set $\{s_k, r_k\}$ together with the input A. The computed positive definite solution is denoted by $\tilde{A}$.

Because of round-off errors, a computed solution $\tilde{A}$ does not satisfy $\tilde{A}_k r_k = s_k r_k$, $k = 1, \ldots, n$, exactly, where $\tilde{A}_k$ is the kth leading principle submatrix of $\tilde{A}$. We use ϕ defined by

$$\phi(\tilde{A}) = \frac{1}{n}\sum_{k=1}^{n} \|\tilde{A}_k r_k - s_k r_k\|_2$$

to measure the reconstruction error of the computed solution $\tilde{A}$. (We did not use the relative error $\psi(\tilde{A}) = \frac{1}{n}\sum_{k=1}^{n} \|\tilde{A}_k r_k - s_k r_k\|_2 / \|\tilde{A}_k\|_2$ because $\tilde{A}_k$ may be zero, though such a zero $\tilde{A}_k$ never occurs for all our testings and $\psi(\tilde{A})$ is always small. For example, for the positive definite case, the minimum of $\psi(\tilde{A})$ is less than $10^{-14.5}$.)

On the other hand, the positiveness of $\tilde{A}$ is verified by checking the positiveness of the smallest eigenvalue $\lambda_{\min}(\tilde{A})$ of $\tilde{A}$ with machine accuracy. If $\lambda_{\min}(\tilde{A}) \approx O(\epsilon)$, where ϵ is the machine accuracy, $\tilde{A}$ can be referred to as a singular matrix approximately.

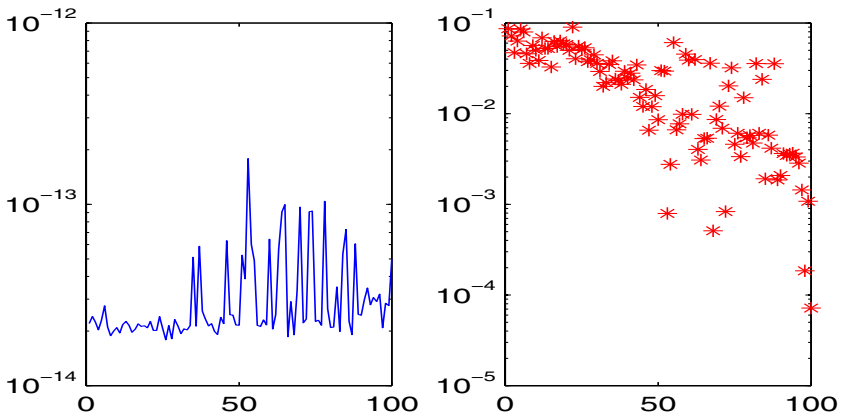


FIG. 2. *Reconstruction error* $\Phi(\tilde{A})$ *(left) and the smallest eigenvalue* $\lambda_{\min}(\tilde{A})$ *(right) of the* 100 *testings.*

The relative measure $\lambda_{\min}(\tilde{A})/\|\tilde{A}\|_2$ can also be used for this purpose.

We experiment with 100 test data sets $\{s_k, r_k\}$ and corresponding A's of order $n = 100$. For each testing, the smallest diagonal r_{ii} of R is smaller than 10^{-14} in absolute value, while the input A is indefinite with the smallest eigenvalue $\lambda_{\min}(A)$ less than -0.1. For the data sets we constructed, all the computed positive definite solutions have high accuracy; see the left plot in Figure 2 for the reconstruction errors $\phi(\tilde{A})$. The smallest eigenvalues $\lambda_{\min}(\tilde{A})$ of the 100 computed solutions $\tilde{A}$'s are plotted in the right plot of Figure 2. It shows that all the computed solutions are positive definite. In general, the norm of the computed positive definite $\tilde{A}$ is larger than the norm of the input A.

To show the efficiency of our algorithm for the positive semidefiniteness case, we also construct data $\{s_k, r_k\}$ as above, but the generating vector `lambda` of eigenvalues of B is replaced by

```
lambda = rand(n,1); lambda(1:3) = 0;
```

Thus the problem SRIEP has at least a positive semidefinite solution. We also update each B to obtain an indefinite solution $A = B + Z$ as before. Totally, 100 numerical examples with $n = 100$ are tested.

Different formulae are used to construct positive semidefinite solutions, depending on whether or not AQ_{01} is zero. In the case when $AQ_{01} = 0$, the algorithm computes a positive semidefinite solution by the formula that is almost the same as that for a positive definite solution. The computed solutions hence have accuracy as good as that in the positive definite case. If AQ_{01} is not zero, it is required to update A by solving the overdetermined system (5.6). This step may reduce the accuracy of computed solutions. However, the computed solutions also have acceptable accuracy in our testings. In Figure 3, we plot the sorted reconstruction errors $\phi(\tilde{A})$, marked by small starts and circles to distinguish the solutions computed with or without solving a corresponding linear system, respectively. Among the 100 testings, 21 ones require updating by solving a relative small linear system (5.6) which size is variable, depending on the inputs. On the left-hand side of Table 1, we list the minimum, mean, and maximum of the numbers of equations in the reduced system, respectively.

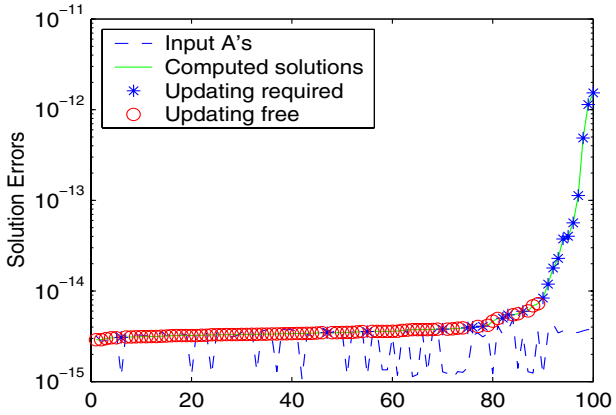


FIG. 3. *Solution errors for the* 100 *testings.*

TABLE 1
Size of the linear systems (5.1) *and* (5.6).

(5.6)	min	mean	max	max	mean	min	(5.1)
# of equ.	297	400.71	594	10197	10079	9999	# of equ.
# of var.	67	122.71	212	4950	4950	4950	# of var.

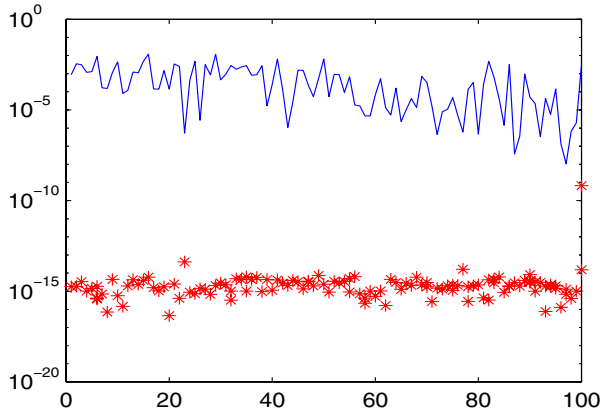


FIG. 4. *Semidefiniteness curve* $\mu(\tilde{A})$ *(solid line) and some smallest eigenvalues of* $\tilde{A}$ *(*) for the* 100 *testings.*

As a comparison, we also list the size of the unreduced system (5.1) on the right. Clearly, the system size is reduced substantially.

To check semidefiniteness of the computed $\tilde{A}$, we use $\mu(\tilde{A})$ defined by

$$\mu(\tilde{A}) = \min\{\, \lambda \;\big|\; \lambda \in \lambda(\tilde{A}) \text{ and } |\lambda| > 10^{-14}\|\tilde{A}\|_2 \,\},$$

where $\lambda(\tilde{A})$ denotes the set of the eigenvalues of $\tilde{A}$. In Figure 4, we plot $\mu(\tilde{A})$ for the 100 testings in the same testing order as in Figure 3. Eigenvalues of $\tilde{A}$ satisfying $|\lambda| \le 10^{-14}\|\tilde{A}\|_2$ are also plotted and marked by $*$. Figure 4 shows that the computed solutions are numerically positive semidefinite.

7. Conclusions. The difficult question of existence of a positive definite or semidefinite solution to the SRIEP when the recursive matrix R is singular is completely answered in this paper. We have completely characterized the necessary and sufficient conditions of existence for such a solution. Indeed, when an (indefinite) solution A to the SRIEP is known, we formulated a subset of positive definite/semidefinite solutions in terms of A, orthogonal basis matrices of the null space of R, and the null space of R^T. By these formulae, it is simple to construct positive solutions or positive semidefinite solutions when the rank quality $\text{rank}(S \circ R^T R) = \text{rank}(AR)$ holds. If the rank constraint is not satisfied, it is required to update A to satisfy the rank constraint. This updating needs a solution of an (overdetermined) linear system. Thus our algorithm proposed in this paper requires a priori an arbitrary solution A to the SRIEP to begin with. Finding an (indefinite) solution to the SRIEP when R is singular is still an open problem. We haven't touched upon the sensitivity analysis of the SRIEP or the error analysis for solving the overdetermined linear system yet. These issues deserve further investigation.

Acknowledgments. We want to thank the anonymous referees and Prof. Moody Chu for their careful reading of this paper. Their insightful suggestions and comments have greatly improved the presentation of this paper.

REFERENCES

[1] M. Arav, D. Hershkowitz, V. Mehrmann, and H. Schneider, *The recursive inverse eigenvalue problem*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 392–412.

[2] W. Givens, *Computation of plane unitary rotations transforming a general matrix to triangular form*, J. Soc. Indust. Appl. Math., 6 (1958), pp. 26–50.

[3] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[4] R. Loewy and V. Mehrmann, *A note on the symmetric recursive inverse eigenvalue problem*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 180–187.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 4, pp. 1132–1149

A METHOD FOR GENERATING INFINITE POSITIVE SELF-ADJOINT TEST MATRICES AND RIESZ BASES*

C. V. M. VAN DER MEE† AND S. SEATZU†

Dedicated to Laura Gori on the occasion of her 70th birthday

Abstract. In this article we propose a method to easily generate infinite multi-index positive definite self-adjoint matrices as well as Riesz bases in suitable subspaces of $L^2(\mathbb{R}^d)$. The method is then applied to obtain some classes of multi-index Toeplitz matrices which are bounded and strictly positive on $\ell^2(\mathbb{Z}^d)$. The condition number of some of these matrices is also computed.

Key words. test matrix, nonuniform sampling, shift-invariant subspace, Riesz basis

AMS subject classifications. 42C15, 46A35

DOI. 10.1137/S0895479803432502

1. Introduction.

Matrices with special properties are important tools for testing numerical algorithms and software, while Riesz bases in different Hilbert spaces are important for solving many problems in approximation theory. However, whereas there are several methods for generating extensive classes of finite test matrices (see, e.g., [16, 11]), we are not aware of methods for generating multi-index test matrices. Similarly, whereas there are methods for generating Riesz bases in subspaces of $L^2(\mathbb{R})$ and $L^2(\mathbb{R}^+)$ [13, 18], we are not aware of general methods for generating Riesz bases in subspaces of $L^2(\mathbb{R}^d)$ for $d \geq 2$, except for grids of sampling points with, apart from a positive constant factor, only integer coordinates [20, 2]. We note that there is an increasing interest in this topic both from the theoretical and the applicational points of view. Classes of multi-index positive definite test matrices could be used, in particular, to compare the effectiveness of preconditioning techniques in solving linear systems by the conjugate gradient method [22, 9, 23, 24].

In a recent joint paper [18] with Nashed on the sampling expansions of functions defined on the real line which belong to unitarily translation invariant reproducing kernel Hilbert spaces H_ϕ, we have developed a method to generate both infinite positive self-adjoint matrices and Riesz bases in suitable subspaces of H_ϕ. More precisely, starting from a real function $\phi \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ whose Fourier transform $\hat{\phi}$ defined by $\hat{\phi}(\omega) = (2\pi)^{-d/2} \int_{\mathbb{R}^d} e^{i\omega\cdot x}\phi(x)dx$ does not vanish, we have represented the Hilbert space H_ϕ of all f such that $(\hat{f}/\hat{\phi}) \in L^2(\mathbb{R})$ as a reproducing kernel Hilbert space with reproducing kernel

$$k_\phi(t,u) = \kappa_\phi(t-u) = \int_{-\infty}^{\infty} \phi(x-t)\phi(x-u)\,dx. \tag{1.1}$$

Assuming in addition that $\phi(\cdot)(1+(\cdot)^2)^\gamma \in L^2(\mathbb{R})$ for some $\gamma > 1$, and taking a sequence of sampling points $\{t_j\}_{j=-\infty}^{\infty}$ such that $|t_i - t_j| \geq \varepsilon > 0$ for $i \neq j$, it has been

*Received by the editors July 8, 2003; accepted for publication (in revised form) by H. J. Woerdeman August 27, 2004; published electronically May 6, 2005. The research leading to this article was supported in part by MIUR under the COFIN grant 2002014121 and by INdAM-GNCS.
http://www.siam.org/journals/simax/26-4/43250.html
†Dipartimento di Matematica e Informatica, Università di Cagliari, viale Merello 92, 09123 Cagliari, Italy (cornelis@bugs.unica.it, seatzu@unica.it).

proved that for all f in a suitable closed subspace $\mathcal{X}_\phi$ of H_ϕ we have the following results:

(a) The *Gram matrix*

$$G_{ij} = k_\phi(t_i, t_j), \quad i, j \in \mathbb{Z},$$

is bounded and strictly positive self-adjoint on $\ell^2(\mathbb{Z})$.

(b) The *sequence*

$$\{k_\phi(\cdot, t_j)\}_{j=-\infty}^{\infty}$$

is a Riesz basis in $\mathcal{X}_\phi$.

(c) The *sampling expansion*

$$f(t) = \frac{1}{\|\phi\|_2^2} \sum_{j=-\infty}^{\infty} f(t_j)\kappa_\phi(t - t_j), \quad t \in \mathbb{R}, \tag{1.2}$$

is valid for every $f \in \mathcal{X}_\phi$. Note that $\kappa_\phi(0) = \|\phi\|_2^2$.

Though the closure $\mathcal{X}_\phi$ of the linear span of the functions $\{k_\phi(\cdot, t_j)\}_{j=-\infty}^{\infty}$ has not been explicitly specified, in [18] various examples have been worked out in detail.

In [18] the main emphasis of the research has been on the development of sampling expansions in unitarily translation invariant reproducing kernel Hilbert spaces. Although in the present article we have generalized the main results in [18] on sampling expansions for functions on the line to sampling expansions for functions on $\mathbb{R}^d$, the present authors are primarily interested in the multi-index Toeplitz matrices arising as Gram matrices of the Riesz bases involved in the case of equidistant sampling points. These matrices are presently being used as test matrices in the development of numerical methods for solving multi-index Toeplitz systems. We are, in particular, interested in comparing the effectiveness of recent preconditioning techniques in solving linear systems by the conjugate gradient method with the most commonly used preconditioning techniques. We are also interested in solving large multi-index Toeplitz systems by using the solution of the corresponding infinite Toeplitz system. For these reasons the present paper contains many explicit examples whose entries have Gaussian, exponential, or algebraic decay away from the diagonal, including the condition numbers of some of the Toeplitz matrices generated.

The outline of the paper is as follows. In section 2 we compile some useful definitions and results involving Gram matrices, Riesz bases, and frame inequalities. In section 3 we illustrate the method proposed for generating positive definite multi-index Toeplitz matrices. In section 4 we present various examples of strictly positive self-adjoint multi-index Toeplitz matrices. Finally, in Appendix A we present a duplication formula for Bessel polynomials that has been used to generate a specific class of multi-index Toeplitz matrices, while in Appendix B we compute the condition numbers of some of the Toeplitz matrices introduced.

Throughout this article, $|\cdot|$ will stand for the Euclidean vector norm or the absolute value of a real or complex number.

2. Preliminaries. Given a complex Hilbert space H, a sequence $\{f_n\}_{n\in J}$, $J \subseteq \mathbb{Z}^d$ and J infinite, of vectors in H is called a *frame* (cf. [10, 26]) if there exist positive constants C_1, C_2 such that

$$C_1\|f\|_H \le \left[\sum_{n\in J} |\langle f, f_n\rangle_H|^2\right]^{1/2} \le C_2\|f\|_H, \quad f \in H.$$

These inequalities are called the *frame inequalities*. The frame is called an *exact frame* if the removal of any vector from the frame causes it not to be a frame anymore. Given a frame, the linear operator T defined by $Tf = \sum_{n\in J}\langle f, f_n\rangle_H f_n$ is a bounded linear operator on H. Further, if $\{f_n\}_{n\in J}$ is an exact frame, for every $f \in H$ there exists a unique sequence $\{a_n\}_{n\in J}$ such that

$$f = \sum_{n\in J} a_n f_n,$$

where $\sum_{n\in J}|a_n|^2 < \infty$. A well-known result [10, 26] states that a sequence $\{f_n\}_{n\in J}$ in a separable Hilbert space H is an exact frame if and only if it is a Riesz basis in H (i.e., if it can be obtained from an orthonormal basis in H by applying a boundedly invertible operator).

PROPOSITION 2.1. *Let H be a complex Hilbert space and let $\{f_j\}_{j\in J}$, $J \subseteq \mathbb{Z}^d$, be a sequence of functions in H. Then the following statements are equivalent:*

1. *There exist positive constants C_1, C_2 such that*

$$C_1\|f\|_H \le \left[\sum_{j\in J}|\langle f, f_j\rangle_H|^2\right]^{1/2} \le C_2\|f\|_H, \quad f \in H, \tag{2.1}$$

holds for every $f \in H$ and no such relation holds for any proper subset of functions $\{f_j\}$.

2. *The sequence $\{f_j\}_{j\in J}$ is a Riesz basis in H.*

3. *The sequence of functions $\{f_j\}_{j\in J}$ is complete, and the Gram matrix $G_{ij} = (\langle f_i, f_j\rangle_H)_{i,j\in J}$ is bounded and strictly positive self-adjoint on $\ell^2(J)$.*

Recall that by a *reproducing kernel Hilbert space* of functions supported on a set S we mean a (complex) Hilbert space of functions on S, where all of the evaluation functionals $\xi_t(f) = f(t)$, for $f \in H$ and each fixed $t \in S$, are continuous [3, 5, 17]. Then, by the Riesz representation theorem, for each $t \in S$ there exists a unique element $k_t \in H$ such that

$$f(t) = \langle f, k_t\rangle, \quad f \in H,$$

where $\langle\cdot,\cdot\rangle$ is the scalar product on H. We then call $k(t,u) = \langle k_t, k_u\rangle$, for $t, u \in S$, the *reproducing kernel* of H. Clearly, $k(\cdot,\cdot)$ is Hermitian and positive definite.

In [18], Proposition 2.1 has been applied more specifically to the situation in which H is a reproducing kernel Hilbert space of complex-valued functions on a set S with reproducing kernel $k(t,s)$ and $f_j(t) = k(t,t_j)/\sqrt{k(t_j,t_j)}$ for a sequence of points $\{t_j\}_{j\in J}$ in S. Then, under any of the conditions of Proposition 2.1, for every $f \in H$ we have the moment expansion

$$f(t) = \sum_{j\in J}\langle f, f_j\rangle_H f_j(t). \tag{2.2}$$

When $J = \mathbb{Z}^d$, the Gram matrix $\{G_{ij}\}_{i,j\in\mathbb{Z}^d}$ is a multi-index Toeplitz matrix (i.e., $G_{ij} = G_{i-j}$ for $i, j \in \mathbb{Z}^d$).

The following elementary result has been adapted from [18].

PROPOSITION 2.2. *Let $J = \mathbb{Z}^d$. Then the statements of Proposition* 2.1 *and the following two claims are equivalent:*

1. *The sequence of functions* $\{f_j\}_{j\in\mathbb{Z}^d}$ *is complete, and the multi-index Toeplitz matrix* $(G_{i-j})_{i,j\in\mathbb{Z}^d}$ *defined by*

$$G_{i-j} = \langle f_i, f_j\rangle_H$$

is bounded and strictly positive self-adjoint on $\ell^2(\mathbb{Z}^d)$.

2. *The sequence of functions* $\{f_j\}_{j\in\mathbb{Z}^d}$ *is complete, and the symbol*

$$\hat{G}(s) = \sum_{j\in\mathbb{Z}^d} s^j G_j, \quad s = (s_1,\dots,s_d), \quad |s_1| = \cdots = |s_d| = 1,$$

is positive, essentially bounded, and essentially bounded away from zero.
If any of these conditions holds and $J = \mathbb{Z}^d$, *the condition number of* G *equals*

$$\frac{\max_{s\in\mathbb{T}^d}\hat{G}(s)}{\min_{s\in\mathbb{T}^d}\hat{G}(s)}, \tag{2.3}$$

where $\mathbb{T}^d$ *is the* d*-dimensional torus.*

3. The method. Let ϕ be a real function in $L^1(\mathbb{R}^d)\cap L^2(\mathbb{R}^d)$ and let

$$k_\phi(t,u) = \kappa_\phi(t-u) = \int_{\mathbb{R}^d}\phi(x-t)\phi(x-u)\,dx = \int_{\mathbb{R}^d} e^{-i\omega(t-u)}|\hat{\phi}(\omega)|^2\,d\omega. \tag{3.1}$$

Then

$$\hat{\kappa}_\phi(\omega) = (2\pi)^{d/2}|\hat{\phi}(\omega)|^2, \tag{3.2}$$

where $\hat{\phi}(\omega) = (2\pi)^{-d/2}\int_{\mathbb{R}^d} e^{i\omega\cdot x}\phi(x)\,dx$. Now suppose $\{t_j\}_{j\in J}$, $J\subset\mathbb{Z}^d$, is an infinite sequence of sampling points in $\mathbb{R}^d$ and

$$(G_\phi)_{ij} := k_\phi(t_i,t_j) = \int_{\mathbb{R}^d}\phi(x-t_i)\phi(x-t_j)\,dx, \quad i,j\in J,$$

the associated Gram matrix. When the sampling points are equidistant (i.e., when $t_j = \alpha j$ for some $\alpha>0$), G_ϕ is a multi-index Toeplitz matrix whose symbol we define by

$$\hat{G}(s,\alpha) = \sum_{j\in\mathbb{Z}^d} s^j\int_{\mathbb{R}^d}\phi(x)\phi(x+\alpha j)\,dx = \sum_{j\in\mathbb{Z}^d} s^j\kappa_\phi(\alpha j),$$

where the series converge uniformly and absolutely in s on the d-dimensional torus $\mathbb{T}^d$ if the condition

$$\sum_{j\in\mathbb{Z}^d}|\kappa_\phi(\alpha j)| < \infty \tag{3.3}$$

is satisfied.

The condition that $\hat{\phi}(\omega)\neq 0$ for every $\omega\in\mathbb{R}^d$ is sufficient for $k_\phi(\cdot,\cdot)$ to be a reproducing kernel on $S=\mathbb{R}^d$. Indeed, let $t_1,\dots,t_n$ be distinct points in $\mathbb{R}^d$. Then for every nontrivial n-tuple $(\xi_1,\dots,\xi_n)$ of complex numbers we have

$$\begin{aligned}\sum_{i,j=1}^n k_\phi(t_i,t_j)\xi_i\overline{\xi_j} &= \int_{\mathbb{R}^d}|\hat{\phi}(\omega)|^2\sum_{i,j=1}^n e^{i(t_i-t_j)\cdot\omega}\xi_i\overline{\xi_j}\,d\omega\\ &= \int_{\mathbb{R}^d}|\hat{\phi}(\omega)|^2\left|\sum_{i=1}^n e^{it_i\cdot\omega}\xi_i\right|^2 d\omega > 0,\end{aligned}$$

which proves that $k_\phi(\cdot,\cdot)$ is a reproducing kernel on $S = \mathbb{R}^d$ if $\hat{\phi}(\omega) \neq 0$ for every $\omega \in \mathbb{R}^d$. As in [18], we now easily identify the corresponding reproducing kernel Hilbert space H_ϕ (cf. [3, 5, 17] for reproducing kernel Hilbert spaces) with the complex Hilbert space of all measurable functions f on $\mathbb{R}^d$ such that $(\hat{f}/\hat{\phi}) \in L^2(\mathbb{R}^d)$, endowed with the norm

$$\|f\|_{H_\phi} = \frac{1}{(2\pi)^{d/2}} \left[\int_{\mathbb{R}^d} |\hat{f}(\omega)|^2 \frac{d\omega}{|\hat{\phi}(\omega)|^2} \right]^{1/2}.$$

The following result provides a general condition on ϕ and the sampling points in order that the Gram matrix $\{\kappa_\phi(t_i, t_j)\}_{i,j\in J}$ be bounded on $\ell^2(J)$. In the case of equidistant sampling points, we actually prove that condition (3.3) holds. Note that all of the examples given in the next section satisfy these conditions.

THEOREM 3.1. *Let the distinct sampling points $\{t_j\}_{j\in J}$, with $J \subseteq \mathbb{Z}^d$ an infinite set, satisfy $|t_i - t_j| \geq \varepsilon > 0$ for $i \neq j$ in J. Further, let ϕ have the property*

$$\exists \gamma > d : \int_{\mathbb{R}^d} (1 + |x|^2)^\gamma \phi(x)^2 \, dx < \infty. \tag{3.4}$$

Then the Gram matrix $\{k_\phi(t_i, t_j)\}_{i,j\in J}$ is bounded on $\ell^2(J)$. In particular, if $t_i = \alpha i$ $(i \in J = \mathbb{Z}^d)$ for some $\alpha > 0$, then (3.3) is satisfied.

Proof. Note that

$$\sup_{i\in J} \sum_{j\in J} |k_\phi(t_i, t_j)| = \sup_{i\in J} \sum_{j\in J} |\kappa_\phi(t_i - t_j)| \tag{3.5}$$

is an upper bound for the norm of the Gram matrix on $\ell^2(J)$. Therefore,

$$\begin{aligned}(1 + |t|)^\gamma |\kappa_\phi(t)| &\leq \int_{\mathbb{R}^d} (1 + |x|)^\gamma |\phi(x)| \cdot (1 + |x + t|)^\gamma |\phi(x + t)| \, dx \\ &\leq \int_{\mathbb{R}^d} (1 + |x|)^{2\gamma} \phi(x)^2 \, dx \leq 2^\gamma \int_{\mathbb{R}^d} (1 + |x|^2)^\gamma \phi(x)^2 \, dx,\end{aligned}$$

which implies that (3.5) is bounded above when $|t_i - t_j| \geq \varepsilon$ for $i \neq j$. ∎

We now give sufficient conditions on ϕ and the sampling points for the Gram matrix $\{\kappa_\phi(t_i, t_j)\}_{i,j\in J}$ to be bounded below on $\ell^2(J)$ by a positive multiple of the identity. With Theorem 3.1, we then obtain sufficient conditions on ϕ and the sampling points in order that this Gram matrix be bounded and strictly positive self-adjoint on $\ell^2(J)$ and that the frame inequalities (2.1) be satisfied. All of the examples of the next section satisfy these conditions. The two proofs we give are based in part on ideas of Schaback [21, Theorem 3.1].

THEOREM 3.2. *Let $(t_j)_{j\in\mathbb{Z}^d}$ be sampling points with $t_0 = 0$ and*

$$|t_{is} - t_{js}| \geq \varepsilon |i_s - j_s| > 0, \quad i, j \in \mathbb{Z}^d \text{ with } t_{is} \neq t_{js}.$$

Let ϕ be a real function in $L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ satisfying the conditions of Theorem 3.1 whose Fourier transform $\hat{\phi}(\omega) \neq 0$ for $\max(|\omega_1|, \ldots, |\omega_d|) \leq M$ for any $M > \pi/\varepsilon\sqrt{3(2^{1/d} - 1)}$. Then the Gram matrix $\{\kappa_\phi(t_i - t_j)\}_{i,j\in\mathbb{Z}^d}$ is bounded and strictly positive self-adjoint on $\ell^2(\mathbb{Z}^d)$.

Moreover, if $\hat{\phi}(\omega) \neq 0$ *for all* $\omega \in \mathbb{R}^d$ *and* $\mathcal{X}_\phi$ *denotes the closed linear span of* $\kappa_{\phi_j}(x) = \kappa_\phi(x - t_j)$, $j \in \mathbb{Z}^d$ *in* H_ϕ, *then there exist positive constants* C_1, C_2 *such that the frame inequalities*

$$C_1\|f\|_{H_\phi} \leq \left[\sum_{j\in\mathbb{Z}^d} |f(t_j)|^2\right]^{1/2} \leq C_2\|f\|_{H_\phi}, \quad f \in \mathcal{X}_\phi, \tag{3.6}$$

hold. Consequently, $\{\kappa_{\phi_j}\}_{j\in\mathbb{Z}^d}$ *is a Riesz basis in* $\mathcal{X}_\phi$ *and for each* $f \in \mathcal{X}_\phi$ *we have the interpolating expansion*

$$f(t) = \frac{1}{\|\phi\|_2^2} \sum_{j\in\mathbb{Z}^d} f(t_j)\kappa_\phi(t - t_j). \tag{3.7}$$

Proof. We present two proofs, the first one adapted to ϕ such that $\hat{\phi}(\omega)$ is zero free for $\max(|\omega_1|, \ldots, |\omega_d|) \leq 2M$, and the second one adapted to ϕ such that $\hat{\phi}(\omega)$ is zero free for $|\omega| \leq 2R$, where M and R are specified in the first and second proofs, respectively.

First proof. For $N \in \mathbb{N}$ and any set of N sampling points and arbitrary complex numbers $c_1, \ldots, c_N$, by Parseval's theorem we have

$$\sum_{j,r=1}^{N} c_j\overline{c_r}\kappa_\phi(t_j - t_r) = \int_{\mathbb{R}^d} \left|\sum_{j=1}^{N} c_j\phi(x - t_j)\right|^2 dx = \int_{\mathbb{R}^d} \left|\sum_{j=1}^{N} c_j e^{i\omega\cdot t_j}\hat{\phi}(\omega)\right|^2 d\omega$$

$$\geq \left(\inf_{|\omega_s|\leq 2M,\ s=1,\ldots,d} |\hat{\phi}(\omega)|^2\right) \int_{-2M}^{2M} \cdots \int_{-2M}^{2M} \Psi_M(\omega) \sum_{i,j=1}^{N} c_i\overline{c_j} e^{i\omega\cdot(t_i - t_j)}\, d\omega, \tag{3.8}$$

where

$$\Psi_M(\omega) = \begin{cases} (2M)^{-d} \prod_{s=1}^{d} (2M - |\omega_s|), & |\omega_s| \leq 2M,\ s = 1, \ldots, d, \\ 0 & \text{otherwise.} \end{cases}$$

Putting

$$B(u) = \int_{-1}^{1} (1 - |\zeta|)e^{i\zeta u}\, d\zeta = \begin{cases} \left(\dfrac{\sin(\frac{1}{2}u)}{\frac{1}{2}u}\right)^2, & u \neq 0, \\ 1, & u = 0, \end{cases} \tag{3.9}$$

we obtain for $t_j = (t_{j1}, \ldots, t_{jd})$ $(j = 1, \ldots, N)$

$$\int_{\mathbb{R}^d} \left|\sum_{j=1}^{N} c_j\phi(x - t_j)\right|^2 dx \geq \left(\inf_{|\omega_s|\leq 2M,\ s=1,\ldots,d} |\hat{\phi}(\omega)|^2\right) \sum_{i,j=1}^{N} c_i\overline{c_j} A_{ij},$$

where

$$A_{ij} = \prod_{s=1}^{d} B(2M(t_{is} - t_{js})). \tag{3.10}$$

Now choose $\varepsilon > 0$ such that $|t_{is} - t_{js}| \geq \varepsilon |i_s - j_s|$ for $t_{is} \neq t_{js}$. Then in view of (3.9)

$$0 < \prod_{s=1}^{d} B(2M(t_{is} - t_{js})) = \prod_{\substack{s=1 \\ t_{is} \neq t_{js}}}^{d} B(2M(t_{is} - t_{js}))$$
$$\leq \prod_{\substack{s=1,\ldots,d \\ t_{is} \neq t_{js}}} \frac{1}{(M\varepsilon(i_s - j_s))^2} = \prod_{\substack{s=1,\ldots,d \\ i_s \neq j_s}} \frac{1}{(M\varepsilon(i_s - j_s))^2}.$$

We easily prove, by induction on d, that

$$\sum_{j \in \mathbb{Z}^d} \prod_{s=1}^{d} B(2M(t_{is} - t_{js})) \leq 1 + \sum_{\substack{j \in \mathbb{Z}^d \\ j \neq i}} \prod_{s=1}^{d} B(2M(t_{is} - t_{js}))$$
$$\leq 1 + \sum_{\substack{j \in \mathbb{Z}^d \\ j \neq i}} \prod_{\substack{s=1,\ldots,d \\ i_s \neq j_s}} \frac{1}{(M\varepsilon(i_s - j_s))^2}$$
$$\leq 1 + \sum_{j \in \mathbb{Z}^d \setminus \{0\}} \prod_{\substack{s=1,\ldots,d \\ j_s \neq 0}} \frac{1}{(M\varepsilon j_s)^2} \leq (1 + 2S(M\varepsilon))^d,$$

where

$$S(z) = \sum_{i=1}^{\infty} \frac{1}{(zi)^2} = \frac{\pi^2}{6z^2}.$$

Using Gershgorin's theorem [12, Theorem 8.1.3], it appears that the real symmetric matrix $(A_{ij})_{i,j=1}^{N}$ with elements defined by the right-hand side of (3.10) has all of its diagonal elements equal to 1, and hence all of its eigenvalues λ are real and satisfy

$$|1 - \lambda| \leq \max_{i=1,\ldots,N} \sum_{\substack{j=1 \\ j \neq i}}^{N} |A_{ij}|.$$

Thus its eigenvalues can be found in the open interval from $2 - (1 + 2S(M\varepsilon))^d$ to $(1 + 2S(M\varepsilon))^d$ whose endpoints do not depend on N. Thus if $M > \pi/(\varepsilon\sqrt{3}(2^{1/d} - 1)^{1/2})$, this matrix is positive definite. Therefore, for this choice of M the lower bound (3.8) extends to arbitrary subsets of the set of the sampling points, and hence the Gram matrix $\{\kappa_\phi(t_i - t_j)\}_{i,j \in \mathbb{Z}}$ is strictly positive self-adjoint. Its boundedness follows from Theorem 3.1. The frame inequalities (3.6) now follow with the help of Proposition 2.1. Finally, (3.7) is immediate from (2.2), (3.6), and $k_\phi(t_j, t_j) = \kappa_\phi(0) = \|\phi\|_2^2$.

Second proof. Let R be a positive real number and let $\chi_R^d(x)$ be the characteristic function of the sphere in $\mathbb{R}^d$ with center the origin and radius R. Then

$$0 \leq \int_{\mathbb{R}^d} dx \chi_R^d(x - t)\chi_R^d(x - s) \leq R^d V_d, \quad t, s \in \mathbb{R}^d,$$

where V_d is the volume of the unit ball in $\mathbb{R}^d$. Then for $N \in \mathbb{N}$ and any set of N

sampling points and arbitrary complex numbers $c_1, \dots, c_N$, we have

$$\begin{aligned}\sum_{j,r=1}^{N} c_j \overline{c_r} \kappa_\phi(t_j - t_r) &= \int_{\mathbb{R}^d} \left| \sum_{j=1}^{N} c_j \phi(x - t_j) \right|^2 dx = \int_{\mathbb{R}^d} \left| \sum_{j=1}^{N} c_j e^{i\omega \cdot t_j} \hat{\phi}(\omega) \right|^2 d\omega \\ &\geq \left(\inf_{|\omega| \leq 2R} |\hat{\phi}(\omega)|^2 \right) \int_{|\omega| \leq 2R} \Psi_R^d(\omega) \sum_{i,j=1}^{N} c_i \overline{c_j} e^{i\omega \cdot (t_i - t_j)} \, d\omega \\ &\geq \left(\inf_{|\omega| \leq 2R} |\hat{\phi}(\omega)|^2 \right) \sum_{i,j=1}^{N} c_i \overline{c_j} F_d(R, t_i - t_j), \qquad (3.11)\end{aligned}$$

where $\hat{\phi}(\omega)$ is zero free for $|\omega| \leq 2R$,

$$\Psi_R^d(\omega) = \frac{1}{R^d V_d} \int_{\mathbb{R}^d} \chi_R^d(\xi) \chi_R^d(\xi - \omega) \, d\xi,$$

and

$$F_d(R,t) = \int_{|\omega| \leq 2R} \Psi_R^d(\omega) e^{i\omega \cdot t} \, d\omega = (2\pi)^{d/2} \hat{\Psi}_R^d(t) = \frac{(2\pi)^d}{R^d V_d} |\hat{\chi}_R^d(t)|^2.$$

Using [14, 8.411(5) and 6.561(5)] and $S_{d-2} = 2\pi^{(d-1)/2}/\Gamma((d-1)/2)$ we easily compute

$$\begin{aligned}\hat{\chi}_R^d(t) &= (2\pi)^{-d/2} S_{d-2} \int_0^R dr r^{d-1} \int_0^\pi d\varphi_1 (\sin \varphi_1)^{d-2} e^{ir|t| \cos \varphi_1} \\ &= (2\pi)^{-d/2} R^d \int_0^1 d\rho \rho^{d-1} \int_0^\pi (\sin \varphi_1)^{d-2} \cos(\rho R |t| \cos \varphi_1) \\ &= R^2 \left(\frac{R}{|t|} \right)^{\frac{d-2}{2}} \int_0^1 d\rho \rho^{\frac{d}{2}} J_{\frac{d-2}{2}}(\rho R|t|) = \left(\frac{R}{|t|} \right)^{\frac{d}{2}} J_{\frac{d}{2}}(R|t|),\end{aligned}$$

so that

$$F_d(R,t) = \frac{(2\pi)^d}{V_d |t|^d} J_{\frac{d}{2}}(R|t|)^2 \text{ and hence } F_d(R,0) = \frac{(\pi R)^d}{V_d \Gamma(\frac{d+2}{2})^2}.$$

According to [14, 8.479], for $|t| \geq (d/2R)$ we have the estimate

$$F_d(R,t) = \frac{(2\pi)^d}{V_d |t|^d} J_{\frac{d}{2}}(R|t|)^2 \leq \frac{2}{\pi} \frac{(2\pi)^d}{V_d |t|^d} \frac{1}{\sqrt{(R|t|)^2 - (\frac{d}{2})^2}},$$

and hence for $|t| \geq ((\mu d)/(2R))$ with fixed $\mu > 1$ we have

$$F_d(R,t) \leq \frac{1}{\pi V_d R \sqrt{\mu^2 - 1}} \left(\frac{2\pi}{|t|} \right)^{d+1}.$$

Now choose $\varepsilon > 0$ such that $|t_i - t_j| \geq \varepsilon [\sum_{s=1}^d (i_s - j_s)^2]^{1/2}$. Then for $\varepsilon \geq ((\mu d)/(2R))$ and some $\mu > 1$ we have for $i \neq j$

$$F_d(R, t_i - t_j) \leq \frac{(2\pi/\varepsilon)^{d+1}}{\pi V_d R \sqrt{\mu^2 - 1}} \frac{1}{[\sum_{s=1}^d (i_s - j_s)^2]^{\frac{d+1}{2}}}.$$

Therefore, for $|t| \geq ((\mu d)/(2R))$ with fixed $\mu > 1$ we have

$$\max_{i=1,\ldots,N} \sum_{\substack{j=1 \\ j\neq i}}^{N} F_d(R, t_i - t_j) \leq \frac{(2\pi/\varepsilon)^{d+1}}{\pi V_d R\sqrt{\mu^2-1}} S_{[d]},$$

where $S_{[d]} = \sum_{0\neq j\in\mathbb{Z}^d} [j_1^2 + \cdots + j_d^2]^{-\frac{d+1}{2}}$. Using Gershgorin's theorem [12, Theorem 8.1.3], it appears that the real symmetric matrix $(F(t_i - t_j))_{i,j=1}^N$, appearing in (3.11), has its eigenvalues in the open interval of half-length $(2\pi/\varepsilon)^{d+1} S_{[d]}/\pi V_d R\sqrt{\mu^2-1}$ centered about $F_d(R, 0)$. Consequently, if ε strictly exceeds the number $\varepsilon_0(R, d, \mu)$ defined by

$$\varepsilon_0(R, d, \mu) = \max\left(\frac{\mu d}{2R}, \frac{2}{R}\left[\frac{S_{[d]}\Gamma(\frac{d+2}{2})^2}{\sqrt{\mu^2-1}}\right]^{\frac{1}{d+1}}\right)$$

for some $\mu > 1$, then the real symmetric matrix $(F(t_i - t_j))_{i,j=1}^N$ appearing in (3.11) is positive definite, irrespective of the choice of finite subset of the sampling points. The frame inequalities (3.6) now follow with the help of Proposition 2.1. Finally, (3.7) is immediate from (2.2), (3.6), and $k_\phi(t_j, t_j) = \kappa_\phi(0) = \|\phi\|_2^2$. □

Assuming $\hat{\phi}(\omega) \neq 0$ for all $\omega \in \mathbb{R}^d$ and given the finite linear combination $\sum_j c_j \kappa_\phi(\cdot - t_j)$ in $\mathcal{X}_\phi$, we easily compute that

$$\begin{aligned}\left\|\sum_j c_j \kappa_\phi(\cdot - t_j)\right\|_{H_\phi}^2 &= \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\left|\sum_j c_j e^{i\omega\cdot t_j}\right|^2 |\hat{\kappa}_\phi(\omega)|^2 \frac{d\omega}{|\hat{\phi}(\omega)|^2} \\ &= \int_{\mathbb{R}^d}\left|\sum_j c_j e^{i\omega\cdot t_j}\right|^2 |\hat{\phi}(\omega)|^2\, d\omega.\end{aligned}$$

Hence if $\boldsymbol{t} = \{t_j : j \in \mathbb{Z}^d\}$ denotes the set of sampling points, then the image $\mathcal{F}[\mathcal{X}_\phi]$ of $\mathcal{X}_\phi$ under the Fourier transformation $\mathcal{F}$ coincides with the completion $\mathcal{AP}_{\boldsymbol{t},\phi}$ of the vector space of d-variate almost periodic polynomials with spectrum within $\boldsymbol{t}$ with respect to the scalar product

$$(f, g)_{\mathcal{X}_\phi} = \int_{\mathbb{R}^d} f(\omega)\overline{g(\omega)}|\hat{\phi}(\omega)|^2\, d\omega.$$

Here by the spectrum of a d-variate almost periodic function $f : \mathbb{R}^d \to \mathbb{C}$ we mean the set of all $t \in \mathbb{R}^n$ for which $\lim_{T\to+\infty} \frac{1}{T^d}\int_0^T \cdots \int_0^T e^{-ix\cdot t} f(x)dx \neq 0$, where we note (cf. [6]) that

$$\lim_{T\to+\infty} \frac{1}{T^d}\int_0^T \cdots \int_0^T e^{ix\cdot(u-t)}\, dx_1 \ldots dx_d = \begin{cases} 1, & t = u \in \mathbb{R}^d, \\ 0, & t, u \in \mathbb{R}^d \text{ and } t \neq u. \end{cases}$$

Denoting the Banach space of d-variate almost periodic functions with spectrum within $\boldsymbol{t}$ with respect to the supremum norm by $\mathcal{AP}_{\boldsymbol{t}}$, one can also identify $\mathcal{F}[\mathcal{X}_\phi]$ with the closure of $\hat{\phi}[\mathcal{AP}_{\boldsymbol{t}}]$ in $L^2(\mathbb{R}^d)$. Since $\hat{\phi}[L^\infty(\mathbb{R}^d)]$ is dense in $L^2(\mathbb{R}^d)$ and $\mathcal{AP}_{\boldsymbol{t}}$ is not

dense in $L^\infty[\mathbb{R}^d]$, the space $\mathcal{X}_\phi$ is a proper closed linear subspace of H_ϕ. Furthermore, due to the estimate

$$\left\| \sum_j c_j \kappa_\phi(\cdot - t_j) \right\|_{H_\phi} \le \|\phi\|_2 \sup_{\omega \in \mathbb{R}^d} \left| \sum_j c_j e^{i\omega \cdot t_j} \right|,$$

we see that

$$\mathcal{AP}_{\boldsymbol{t}} \subset \mathcal{AP}_{\boldsymbol{t},\phi} = \mathcal{F}[\mathcal{X}_\phi].$$

When the sampling points form a rectangular grid in $\mathbb{R}^d$ containing the origin (i.e., when there exists $\alpha > 0$ such that $t_j = \alpha j$ for $j \in \mathbb{Z}^d$), the space $\mathcal{AP}_{\boldsymbol{t}}$ coincides with the Banach space of all bounded continuous functions $f : \mathbb{R}^d \to \mathbb{C}$ satisfying $f(\omega + \frac{2j\pi}{\alpha}) = f(\omega)$ for all $\omega \in \mathbb{R}^d$ and $j \in \mathbb{Z}^d$, endowed with the supremum norm.

4. Examples. Let us discuss the following examples of real functions. Here we remark that if $\phi(t)$ depends only on $|t|$, then $\kappa_\phi(t)$ depends only on $|t|$ and $\hat{\phi}(\omega)$ depends only on $|\omega|$. Consequently, expressing the Cartesian coordinates in spherical coordinates by putting $x_1 = r\cos\varphi_1$, $x_2 = r\sin\varphi_1\cos\varphi_2, \ldots,$ $x_{d-1} = r\sin\varphi_1 \ldots \sin\varphi_{d-2}\cos\varphi_{d-1}$, $x_d = r\sin\varphi_1 \ldots \sin\varphi_{d-2}\sin\varphi_{d-1}$, where we have $\varphi_j \in [0,\pi]$ $(j = 1, \ldots, d-2)$ and $\varphi_{d-1} \in [-\pi, \pi]$, with Jacobian

$$J = r^{d-1}(\sin\varphi_1)^{d-2}(\sin\varphi_2)^{d-3} \ldots \sin\varphi_{d-2},$$

we obtain

$$\hat{\phi}(\omega) = (2\pi)^{-d/2} \left(\frac{2}{|\omega|}\right)^{\frac{d-2}{2}} \Gamma\left(\frac{d}{2}\right) S_{d-1} \int_0^\infty r^{d/2} J_{\frac{d-2}{2}}(|\omega| r) \phi(r)\, dr, \tag{4.1}$$

where S_{d-1} is the surface measure of S^{d-1}, $S_{d-1} = S_{d-2} B(\frac{d-1}{2}, \frac{1}{2})$, and $B(p,q)$ and $J_\nu(z)$ stand for the Euler beta function and the Bessel function of order ν, respectively.

Example 4.1. A typical example involves the Gram matrix of the multinomial distribution [19]. Let Σ be a positive definite real $d \times d$ matrix,

$$\phi(x) = \left(\frac{\det\Sigma}{\pi^d}\right)^{1/2} e^{-(\Sigma x, x)} = \pi^{-d/2}(\det\Sigma)^{1/2} \exp\left(-\sum_{i,j=1}^d \Sigma_{ij} x_i x_j\right), \tag{4.2}$$

where $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ and $\kappa_\phi(t,s) = \int_{\mathbb{R}^d} \phi(x-t)\phi(x-s)dx$. Then

$$\kappa_\phi(t) = \int_{\mathbb{R}^d} \phi(x)\phi(x-t)\, dx = (2\pi)^{-d/2}(\det\Sigma)^{1/2} e^{-\frac{1}{2}(\Sigma t, t)}, \tag{4.3}$$

where $t \in \mathbb{R}^d$. In particular, for $t, s \in \mathbb{R}^d$ we have

$$\hat{\phi}(\omega) = (2\pi)^{-d/2} e^{-\frac{1}{4}(\Sigma^{-1}\omega, \omega)} \neq 0$$

in $\mathbb{R}^d$ and

$$k_\phi(t,s) = \int_{\mathbb{R}^d} \phi(x-t)\phi(x-s)\, dx = \kappa_\phi(t-s) = \kappa_\phi(s-t).$$

Hence, if $t_j = \alpha j$, $\alpha > 0$, $j \in \mathbb{Z}^d$, the Toeplitz matrix

$$G_{i-j}(\alpha) = k_\phi(\alpha|i-j|),$$

whose entries have a Gaussian decay away from the diagonal elements, is bounded and strictly positive self-adjoint on $\ell^2(\mathbb{Z}^d)$. Furthermore, as $\hat{\phi}$ has no zeros in $\mathbb{R}^d$, the expansion (3.7) holds with $\|\phi\|_2^2 = (2\pi)^{-d/2}(\det \Sigma)^{1/2}$.

Example 4.2. For $\sigma > 0$, consider $\phi(x) = e^{-\sigma|x|}$, where the length of $x \in \mathbb{R}^d$ is its Euclidean vector norm. Then $\hat{\phi}(\omega)$ depends only on $|\omega|$ and $\kappa_\phi(x)$ depends only on $|x|$. For $d = 2$ we use $\kappa_\phi(t) = k_\phi(-\frac{1}{2}|t|e_1, \frac{1}{2}|t|e_1)$, where $e_1 = (1, 0)$, and apply the transformation $x = \frac{1}{2}|t|(\cosh u \cos v, \sinh u \sin v)$ to elliptical coordinates (u, v) in (3.1) to find

$$\kappa_\phi(t) = \frac{\pi|t|^2}{4} \int_0^\infty \cosh(2u) e^{-\sigma|t|\cosh u}\, du = \frac{\pi|t|^2}{4} K_2(\sigma|t|),$$

where K_2 stands for McDonald's function [14, 8.432(1)].

For $d \geq 3$ we observe that (1) $\kappa_\phi(t) = k_\phi(-\frac{1}{2}|t|e_1, \frac{1}{2}|t|e_1)$, where $e_1 = (1, 0, \ldots, 0)$, and (2) the integrand does not change if the relative position of x in the two-dimensional plane containing $\pm\frac{1}{2}|t|$ and x remains the same. Denoting the surface measure of S^{d-2} by S_{d-2} and using the fact that $S_{d-1} = S_{d-2}B(\frac{d-1}{2}, \frac{1}{2})$, we obtain [1, 9.6.23 and 9.6.26]

$$\begin{aligned}
\kappa_\phi(t) &= \frac{S_{d-1}}{2^d}|t|^d \int_0^\infty \left[\sinh^d u + \frac{d-1}{d}\sinh^{d-2} u\right] e^{-\sigma|t|\cosh u}\, du \\
&= \frac{\Gamma(\frac{d+1}{2})S_{d-1}}{\sqrt{\pi}} \left(\frac{|t|}{2\sigma}\right)^{d/2} \left[K_{\frac{d}{2}}(\sigma|t|) + \frac{\sigma|t|}{d} K_{\frac{d-2}{2}}(\sigma|t|)\right] \\
&= \frac{\sigma^2\Gamma(\frac{d+1}{2})S_{d-1}}{\sqrt{\pi}} \left(\frac{|t|}{2\sigma}\right)^{\frac{d+2}{2}} K_{\frac{d+2}{2}}(\sigma|t|).
\end{aligned}$$

For $d = 3$, in particular, we have

$$\kappa_\phi(t) = \frac{\pi}{2}|t|^3 \int_1^\infty \left(\xi^2 - \frac{1}{3}\right) e^{-\sigma|t|\xi}\, d\xi = \frac{\pi}{\sigma^3}\left(\frac{2}{3}\sigma^2|t|^2 + \sigma|t| + 1\right) e^{-\sigma|t|}.$$

Moreover, for any $d \geq 2$ we have

$$\begin{aligned}
\hat{\phi}(\omega) &= (2\pi)^{-d/2} S_{d-2} \int_0^\infty \int_0^\pi r^{d-1} \sin^{d-2}\theta\, e^{i|\omega|r\cos\theta} e^{-\sigma r}\, d\theta dr \\
&= (2\pi)^{-d/2} S_{d-2} \Gamma\left(\frac{d-1}{2}\right)\Gamma\left(\frac{1}{2}\right)\left(\frac{|\omega|}{2}\right)^{-\frac{d-2}{2}} \int_0^\infty r^{d/2} J_{\frac{d-2}{2}}(|\omega|r) e^{-\sigma r}\, dr \\
&= (2\pi)^{-d/2} S_{d-2} B\left(\frac{d-1}{2}, \frac{1}{2}\right)\Gamma(d) \frac{F\left(\frac{d}{2}, -\frac{1}{2}; \frac{d}{2}; \frac{|\omega|^2}{\sigma^2+|\omega|^2}\right)}{(\sigma^2 + |\omega|^2)^{d/2}} \\
&= (2\pi)^{-d/2} S_{d-2} B\left(\frac{d-1}{2}, \frac{1}{2}\right) \frac{(d-1)!\sigma}{(\sigma^2 + |\omega|^2)^{\frac{d+1}{2}}} \\
&= (2\pi)^{-d/2} S_{d-1} \frac{(d-1)!\sigma}{(\sigma^2 + |\omega|^2)^{\frac{d+1}{2}}},
\end{aligned}$$

where we have used (4.1), [14, line 3 of 6.621(1)], and $S_{d-1} = S_{d-2}B(\frac{d-1}{2}, \frac{1}{2})$, while F stands for the hypergeometric function. For $d = 3$ we trivially find

$$\hat{\phi}(\omega) = \frac{4}{\sqrt{2\pi}} \frac{\sigma}{(\sigma^2 + |\omega|^2)^2},$$

which has no zeros in $\mathbb{R}$. As a result, the Gram matrix G given by $G_{ij} = \kappa_\phi(t_i - t_j)$, $i, j \in \mathbb{Z}^d$, whose entries decay exponentially away from the diagonal, is strictly positive definite and the expansion (3.7) holds with k_ϕ defined as above.

Example 4.3. For $d \geq 2$ and $\sigma > 0$ consider the algebraically decaying function $\phi(x) = (\sigma^2 + |x|^2)^{-\frac{d+1}{2}}$. Then ϕ satisfies condition (3.4) for $\gamma \in (d, d+1)$, while (4.1) and [14, 6.565(3)] imply

$$\hat{\phi}(\omega) = (2\pi)^{-d/2} \frac{S_d}{2\sigma} e^{-\sigma|\omega|},$$

where we have employed $S_{d-1}B(\frac{d}{2}, \frac{1}{2}) = S_d$. Thus

$$\hat{\kappa}_\phi(\omega) = (2\pi)^{d/2} |\hat{\phi}(\omega)|^2 = (2\pi)^{-d/2} \frac{S_d^2}{4\sigma^2} e^{-2\sigma|\omega|}.$$

Consequently,

$$\kappa_\phi(t) = \frac{S_d}{\sigma} \frac{1}{(4\sigma^2 + |t|^2)^{\frac{d+1}{2}}}.$$

More generally, for $d \geq 2$, $\sigma > 0$, and $q = 0, 1, \ldots$ consider $\phi(x) = (\sigma^2 + |x|^2)^{-(\frac{d+1}{2}+q)}$. Then (4.1) and [14, 6.565(4)] imply

$$\begin{aligned}
\hat{\phi}(\omega) &= (2\pi)^{-d/2} \left(\frac{|\omega|}{2\sigma}\right)^{q+\frac{1}{2}} \frac{\Gamma(\frac{d}{2}) S_{d-1}}{\Gamma(\frac{d+1}{2} + q)} K_{-(q+\frac{1}{2})}(\sigma|\omega|) \\
&= (2\pi)^{-d/2} \frac{\theta_q(\sigma|\omega|)}{2^{q+1}\sigma^{2q+1}} \frac{\Gamma(\frac{d}{2})\Gamma(\frac{1}{2})}{\Gamma(\frac{d+1}{2} + q)} S_{d-1} e^{-\sigma|\omega|} \\
&= (2\pi)^{-d/2} \frac{\theta_q(\sigma|\omega|) S_d}{2^{q+1}\sigma^{2q+1}(\frac{d+1}{2})_q} e^{-\sigma|\omega|}, \qquad (4.4)
\end{aligned}$$

where we have used the Pochhammer symbol $c_0 = 1$ and $c_s = c(c+1)(c+2)\cdots(c+s-1)$ for $s = 1, 2, \ldots$ and the expression (see [14, 8.486(14) and 8.486(16)], plus induction on q)

$$K_{\pm(q+\frac{1}{2})}(z) = \sqrt{\frac{\pi}{2z}} \frac{\theta_q(z)}{z^q} e^{-z}$$

for the so-called Bessel polynomials $\theta_q(z)$ of degree q which satisfy the recurrence relations (see [14, 8.486(14) and 8.486(10)])

$$\theta_0(z) = 1, \qquad \theta_1(z) = z + 1, \qquad \theta_{q+1}(z) = z[\theta_q(z) - \theta_q'(z)] + (2q+1)\theta_q(z), \qquad (4.5)$$

$$\theta_0(z) = 1, \qquad \theta_1(z) = z + 1, \qquad \theta_{q+1}(z) = (2q+1)\theta_q(z) + z^2\theta_{q-1}(z) \qquad (4.6)$$

and have the explicit form (see [1, 10.2.15] and [15, Chap. 2, (7)–(8)])

$$\theta_q(z) = \sum_{k=0}^{q} \frac{1}{2^{q-k}} \frac{(2q-k)!}{k!(q-k)!} z^k.$$

Using the fact that $\kappa_\phi(t)$ is a linear combination of expressions of the type (4.4) with σ replaced by 2σ as well as (A.1), we obtain

$$\kappa_\phi(t) = \frac{S_d}{(2\sigma)^{2q+1}[d+1]_q} \sum_{s=q}^{2q} \frac{d_{q,s}\sigma^{2(s-q)}[d+2q+1]_{s-q}}{(4\sigma^2+|t|^2)^{\frac{d+1}{2}+s}},$$

where $d_{q,2q-n} = \frac{(2n)!}{2^n n!}\binom{q}{n}$ $(n = 0, 1, \dots, q)$, $[c]_0 = 1$, and $[c]_s = c(c+2)(c+4)\cdots(c+2s-2)$ for $s = 1, 2, \dots$. In particular, for $q = 1$ we find

$$\kappa_\phi(t) = \frac{S_d}{8(d+1)\sigma^3}\left[\frac{1}{(4\sigma^2+|t|^2)^{\frac{d+3}{2}}} + \frac{\sigma^2(d+3)}{(4\sigma^2+|t|^2)^{\frac{d+5}{2}}}\right].$$

Further, for $q = 2$ and $q = 3$ we find

$$\kappa_\phi(t) = \frac{S_d}{32\sigma^5(d+1)(d+3)}\left[\frac{3}{(4\sigma^2+|t|^2)^{\frac{d+5}{2}}} + \frac{2(d+5)\sigma^2}{(4\sigma^2+|t|^2)^{\frac{d+7}{2}}} + \frac{(d+5)(d+7)\sigma^4}{(4\sigma^2+|t|^2)^{\frac{d+9}{2}}}\right]$$

and

$$\begin{aligned}\kappa_\phi(t) = \frac{S_d}{128\sigma^7(d+1)(d+3)(d+5)}\Bigg[&\frac{15}{(4\sigma^2+|t|^2)^{\frac{d+7}{2}}} + \frac{9\sigma^2(d+7)}{(4\sigma^2+|t|^2)^{\frac{d+9}{2}}} \\ &+ \frac{3\sigma^4(d+7)(d+9)}{(4\sigma^2+|t|^2)^{\frac{d+11}{2}}} \\ &+ \frac{\sigma^6(d+7)(d+9)(d+11)}{(4\sigma^2+|t|^2)^{\frac{d+13}{2}}}\Bigg],\end{aligned}$$

respectively. For $q = 4$ and $q = 5$ we obtain

$$\begin{aligned}\kappa_\phi(t) = \frac{S_d}{512\sigma^9(d+1)(d+3)(d+5)(d+7)}\Bigg[&\frac{105}{(4\sigma^2+|t|^2)^{\frac{d+9}{2}}} \\ &+ \frac{60\sigma^2(d+9)}{(4\sigma^2+|t|^2)^{\frac{d+11}{2}}} + \frac{18\sigma^4(d+9)(d+11)}{(4\sigma^2+|t|^2)^{\frac{d+13}{2}}} \\ &+ \frac{4\sigma^6(d+9)(d+11)(d+13)}{(4\sigma^2+|t|^2)^{\frac{d+15}{2}}} + \frac{\sigma^8(d+7)(d+9)(d+11)(d+13)(d+15)}{(4\sigma^2+|t|^2)^{\frac{d+17}{2}}}\Bigg]\end{aligned}$$

and

$$\begin{aligned}\kappa_\phi(t) = \frac{S_d}{2048\sigma^{11}(d+1)(d+3)(d+5)(d+7)(d+9)}\Bigg[&\frac{945}{(4\sigma^2+|t|^2)^{\frac{d+11}{2}}} \\ &+ \frac{525\sigma^2(d+11)}{(4\sigma^2+|t|^2)^{\frac{d+13}{2}}} + \frac{150\sigma^4(d+11)(d+13)}{(4\sigma^2+|t|^2)^{\frac{d+15}{2}}} \\ &+ \frac{30\sigma^6(d+11)(d+13)(d+15)}{(4\sigma^2+|t|^2)^{\frac{d+17}{2}}} + \frac{5\sigma^8(d+11)(d+13)(d+15)(d+17)}{(4\sigma^2+|t|^2)^{\frac{d+19}{2}}} \\ &+ \frac{\sigma^{10}(d+11)(d+13)(d+15)(d+17)(d+19)}{(4\sigma^2+|t|^2)^{\frac{d+21}{2}}}\Bigg],\end{aligned}$$

respectively. In this example the Gram matrix $\{k_\phi(t_i,t_j)\}_{i,j\in\mathbb{Z}^d}$, whose entries decay algebraically away from the diagonal, is strictly positive self-adjoint. Furthermore, the expansion (3.7) holds with k_ϕ as above, as $\hat{\phi}$ does not have zeros in $\mathbb{R}^d$.

Example 4.4. Now consider the box spline

$$\phi(x_1,x_2)=\begin{cases}1-x_2, & 0\le x_1\le x_2\le 1,\\ 1-x_1, & 0\le x_2\le x_1\le 1,\\ 1-x_1+x_2, & 0\le x_1\le 1,\ -1+x_1\le x_2\le 0,\\ \phi(-x_1,-x_2), & -1\le x_1\le 0,\ -1\le x_2\le 1+x_1,\end{cases}$$

and zero elsewhere. Then

$$\begin{aligned}\hat{\phi}(\omega_1,\omega_2)&=\frac{1}{\pi}\frac{\sin(\omega_1)+\sin(\omega_2)-\sin(\omega_1+\omega_2)}{\omega_1\omega_2(\omega_1+\omega_2)}\\ &=\frac{1}{2\pi}\frac{\sin(\frac{1}{2}\omega_1)}{\frac{1}{2}\omega_1}\frac{\sin(\frac{1}{2}\omega_2)}{\frac{1}{2}\omega_2}\frac{\sin(\frac{1}{2}(\omega_1+\omega_2))}{\frac{1}{2}(\omega_1+\omega_2)},\end{aligned}$$

while $\hat{\phi}(0,0)=(1/2\pi)$ and

$$\hat{\phi}(\omega_1,0)=\hat{\phi}(0,\omega_1)=\hat{\phi}(\omega_1,-\omega_1)=\frac{1}{\pi}\frac{1-\cos(\omega_1)}{\omega_1^2}.$$

Thus

$$\hat{\phi}(\omega_1,\omega_2)>0,\quad \max(|\omega_1|,|\omega_2|,|\omega_1+\omega_2|)<2\pi.$$

As a consequence, the Gram matrix $\{k_\phi(t_i,t_j)\}_{i,j\in\mathbb{Z}^d}$ is positive self-adjoint, but the expansion (3.7) is not valid, because $\hat{\phi}(\omega_1,\omega_2)$ has zeros in $\mathbb{R}^d$.

Let us now employ (3.2) to get

$$\hat{\kappa}_\phi(\omega_1,\omega_2)=\frac{4}{\pi}\frac{1-\cos(\omega_1)}{\omega_1^2}\frac{1-\cos(\omega_2)}{\omega_2^2}\frac{1-\cos(\omega_1+\omega_2)}{(\omega_1+\omega_2)^2}. \tag{4.7}$$

Introducing $\psi(x)=1-|x|$ for $-1\le x\le 1$ and $\psi(x)=0$ for $|x|\ge 1$, so that $\hat{\psi}(\omega)=\sqrt{\frac{2}{\pi}}\frac{1-\cos(\omega)}{\omega^2}$, we can write (4.7) in the form

$$\hat{\kappa}_\phi(\omega_1,\omega_2)=\sqrt{2\pi}\hat{\psi}(\omega_1)\hat{\psi}(\omega_2)\hat{\psi}(\omega_1+\omega_2),$$

which implies that

$$\begin{aligned}\kappa_\phi(t_1,t_2)&=\sqrt{2\pi}\int_{-\infty}^{\infty}d\omega_1 e^{-i\omega_1 t_1}\hat{\psi}(\omega_1)\int_{-\infty}^{\infty}dz\psi(t_2-z)e^{i\omega_1 z}\psi(z)\\ &=\int_{-\infty}^{\infty}dz\psi(z)\psi(t_1-z)\psi(t_2-z)\\ &=\begin{cases}S(t_2), & 0\le t_1\le t_2\le 2,\\ S(t_1), & 0\le t_2\le t_1\le 2,\\ S(t_1-t_2), & 0\le t_1\le 2,\ -2+t_1\le t_2\le 0,\\ k_\phi(-t_1,-t_2), & -2\le t_1\le 0,\ -2\le t_2\le 2+t_1,\end{cases}\end{aligned} \tag{4.8}$$

where

$$S(t) = \begin{cases} \frac{1}{12}t^3 + \frac{2}{3}(1-t) - \frac{1}{6}(1-t)^3, & 0 \le t \le 1, \\ \frac{1}{12}(2-t)^4, & 1 \le t \le 2, \end{cases}$$

and zero outside $[-2,2]^2$.

Appendix A. Some expressions involving Bessel polynomials. In this appendix we prove the following result of independent interest.

THEOREM A.1. *We have*

$$2^{2q}\theta_q(z)^2 = \sum_{n=0}^{q} \frac{(2n)!}{2^n n!}\binom{q}{n}\theta_{2q-n}(2z). \tag{A.1}$$

Proof. According to (5.6) of [8] we have the addition formula

$$\theta_q(z+w) = 2^q \sum_{r=0}^{q} (-1)^{q-r} \frac{q!(2r+1)}{(q-r)!(q+r+1)!}(zw)^{q-r}\theta_r(z)\theta_r(w) \tag{A.2}$$

and the inverse addition formula

$$\theta_q(z)\theta_q(w) = \sum_{r=0}^{q} \frac{(q+r)!}{(q-r)!r!} 2^{-r}(zw)^{q-r}\theta_r(z+w), \tag{A.3}$$

which follow from analogous expressions for the Laguerre polynomials [7]. From (A.3) we have the duplication formula

$$2^{2q}\theta_q(z)^2 = \sum_{r=0}^{q} \frac{(q+r)!}{(q-r)!r!} 2^r (2z)^{2(q-r)}\theta_r(2z). \tag{A.4}$$

Using (3.1) and (1.5) of [8], we see that

$$z^{2k}\theta_n(z) = \sum_{s=0}^{k} (-1)^s \binom{k}{s} \frac{(2n+2k+1)!!}{(2n+2k-2s+1)!!}\theta_{n+2k-s}(z), \tag{A.5}$$

which generalizes (4.6). Substituting (A.5) into (A.4) (with $2z$, r, and $q-r$ instead of z, n, and k) we get

$$\begin{aligned} 2^{2q}\theta_q(z)^2 &= \sum_{r=0}^{q} \frac{(q+r)!}{(q-r)!r!} 2^r \sum_{s=0}^{q-r} (-1)^s \binom{q-r}{s} \frac{(2q+1)!!}{(2q-2s+1)!!}\theta_{2q-r-s}(2z) \\ &= \sum_{n=0}^{q}\sum_{s=0}^{n} (-1)^s 2^{n-s} \frac{(q+n-s)!}{s!(n-s)!(q-n)!} \frac{(2q+1)!!}{(2q-2s+1)!!}\theta_{2q-n}(2z) \\ &= \sum_{n=0}^{q} \binom{q}{n} B(q,n)\theta_{2q-n}(2z), \end{aligned}$$

where

$$B(q,n) = \sum_{s=0}^{n}(-1)^s 2^{n-s}\binom{n}{s}\frac{(q+n-s)!}{q!}\frac{(2q+1)!!}{(2q-2s+1)!!}$$
$$= 2^n(n!)\sum_{s=0}^{n}\frac{(-q-\frac{1}{2})_s(q+1)_{n-s}}{s!(n-s)!} = 2^n(n!)\frac{((-q-\frac{1}{2})+(q+1))_n}{n!}$$
$$= 2^n\left(\frac{1}{2}\right)_n = \frac{(2n)!}{2^n n!}. \tag{A.6}$$

In the penultimate equality of (A.6) we have applied a corollary of the Chu–Vandermonde identity derived in Remark 2.2.1 of [4]. □

Appendix B. Condition numbers. In this appendix the condition numbers $\mathrm{cond}(G_\phi)$ of the multi-index Toeplitz matrix G_ϕ are listed in the case $t_i = \alpha i$ ($i \in \mathbb{Z}^d$) for Examples 4.1–4.4 as far we have been able to compute them, in some cases only for $d = 1$.

Example 4.1. For $d = 1$ we have $\phi(x) = e^{-\sigma x^2}$ and

$$\hat{G}(s,\alpha) = \sqrt{\frac{\pi}{2\sigma}}\left(1 + 2\sum_{j=1}^{\infty} e^{-\sigma\alpha^2 j^2/2}\cos(j\theta)\right)$$
$$= \sqrt{\frac{\pi}{2\sigma}}\vartheta_3\left(\frac{1}{2}\theta, e^{-\sigma\alpha^2/2}\right)$$
$$= G(\alpha)\sqrt{\frac{\pi}{2\sigma}}\prod_{j=1}^{\infty}\left\{\left(1+e^{-(j-\frac{1}{2})\sigma\alpha^2}e^{i\theta}\right)\left(1+e^{-(j-\frac{1}{2})\sigma\alpha^2}e^{-i\theta}\right)\right\},$$

where $s = e^{i\theta}$, ϑ_3 denotes a Jacobian Theta function [25, sect. 21.11 and 21.3], and $G(\alpha) = \prod_{j=1}^{\infty}(1 - e^{-j\sigma\alpha^2})$. Consequently,

$$\mathrm{cond}(G_\phi) = \frac{\hat{G}(1,\alpha)}{\hat{G}(-1,\alpha)} = \left(\prod_{j=1}^{\infty}\frac{1+e^{-(j-\frac{1}{2})\sigma\alpha^2}}{1-e^{-(j-\frac{1}{2})\sigma\alpha^2}}\right)^2.$$

Example 4.2. For $d = 1$ we have $\phi(x) = e^{-\sigma|x|}$ and

$$\hat{G}(s,\alpha) = \alpha\frac{p(\alpha\sigma) + q(\alpha\sigma)[s+s^{-1}]}{(1-se^{-\alpha\sigma})^2(1-s^{-1}e^{-\alpha\sigma})^2},$$

where $p(\beta) = \frac{1}{\beta} - 4e^{-2\beta} - \frac{1}{\beta}e^{-4\beta}$ and $q(\beta) = (1+\frac{1}{\beta})e^{-3\beta} + (1-\frac{1}{\beta})e^{-\beta}$. Consequently,

$$\mathrm{cond}(G_\phi) = \frac{\hat{G}(1,\alpha)}{\hat{G}(-1,\alpha)} = \frac{p(\alpha\sigma)+2q(\alpha\sigma)}{p(\alpha\sigma)-2q(\alpha\sigma)}\left(\frac{1+e^{-\alpha\sigma}}{1-e^{-\alpha\sigma}}\right)^4.$$

Example 4.3. For $d = 1$ and $q = 0$ we have $\phi(x) = 1/(\sigma^2 + x^2)$ and

$$\hat{G}(s,\alpha) = \frac{\pi^2}{\alpha}\frac{2}{\pi\sigma^2}\left(\frac{\alpha}{4\sigma} + \frac{2\sigma}{\alpha}\sum_{j=1}^{\infty}\frac{(-1)^j\cos\{j(\pi-\theta)\}}{j^2+(2\sigma/\alpha)^2}\right)$$
$$= \frac{1}{\sigma^3}\frac{e^{2(\pi-\theta)\sigma/\alpha} + e^{-2(\pi-\theta)\sigma/\alpha}}{e^{2\pi\sigma/\alpha} - e^{-2\pi\sigma/\alpha}},$$

where $s = e^{i\theta}$ (cf. [25, Prob. 9 of Chap. IX]). Consequently,

$$\operatorname{cond}(G_\phi) = \frac{\hat{G}(1,\alpha)}{\hat{G}(-1,\alpha)} = \cosh\left(\frac{2\pi\sigma}{\alpha}\right).$$

Example 4.4. We now compute the Toeplitz matrix $G = (G_{i-j})_{i,j\in\mathbb{Z}^2}$ where

$$G_i = \int_{\mathbb{R}^2} \phi(x)\phi(x-i)\,dx, \quad i = (i_1, i_2) \in \mathbb{Z}^2.$$

By using (4.8) it is immediate to obtain

$$G_i = \begin{cases} \dfrac{1}{2}, & i_1 = i_2 = 0, \\ \dfrac{1}{12}, & i \in \{(1,0),(-1,0),(0,1),(0,-1)\}, \\ \dfrac{1}{12}, & i \in \{(1,1),(-1,-1)\}, \\ 0 & \text{elsewhere.} \end{cases}$$

The corresponding symbol is given by

$$\hat{G}(s, \alpha = 1) = \frac{1}{6}\left(3 + \cos\vartheta_1 + \cos\vartheta_2 + \cos(\vartheta_1 + \vartheta_2)\right) > 0,$$

where $s = (e^{i\vartheta_1}, e^{i\vartheta_2})$, from which we immediately have

$$\operatorname{cond}(G) = \frac{\hat{G}(1,1)}{\hat{G}(e^{2\pi i/3}, e^{2\pi i/3})} = 4.$$

Acknowledgments. The authors are greatly indebted to Prof. Mourad Ismail for allowing them to conclude the proof of Theorem A.1 by telling them about the Chu–Vandermonde identity in [4]. The authors also express their gratitude to the referees for their useful comments.

REFERENCES

[1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, Dover, New York, 1964.
[2] A. Aldroubi and K. Gröchenig, *Nonuniform sampling and reconstruction in shift-invariant spaces*, SIAM Rev., 43 (2001), pp. 585–620.
[3] D. Alpay, *The Schur Algorithm, Reproducing Kernel Spaces and System Theory*, SMF/AMS Texts Monogr. 5, AMS, Providence, RI, 2001.
[4] G. E. Andrews, R. Askey, and R. Roy, *Special Functions*, Encyclopedia Math. Appl. 71, Cambridge University Press, Cambridge, New York, 1999.
[5] N. Aronszajn, *Theory of reproducing kernels*, Trans. Amer. Math. Soc., 68 (1950), pp. 337–404.
[6] H. Bohr, *Almost Periodic Functions*, Chelsea, New York, 1947.
[7] J. L. Burchnall and T. W. Chaundy, *Expansions of Appell's double hypergeometric functions* II, Quart. J. Math., Oxford Ser., 12 (1941), pp. 112–128.
[8] L. Carlitz, *A note on the Bessel polynomials*, Duke Math. J., 24 (1957), pp. 151–162.
[9] T. F. Chan, *An optimal circulant preconditioner for Toeplitz systems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 766–771.
[10] R. J. Duffin and A. C. Schaeffer, *A class of nonharmonic Fourier series*, Trans. Amer. Math. Soc., 72 (1952), pp. 341–366.

[11] W. S. Ericksen, *Inverse pairs of test matrices*, ACM Trans. Math. Software, 11 (1985), pp. 302–304.
[12] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1983.
[13] T. N. T. Goodman, C. A. Micchelli, and Z. Shen, *Riesz bases in subspaces of* $L_2(\mathbb{R}_+)$, Constr. Approx., 17 (2000), pp. 39–46.
[14] I. S. Gradshteyn and I. M. Ryzhik, *Tables of Integrals, Series, and Products*, corrected and enlarged ed., Academic Press, New York, 1980.
[15] E. Grosswald, *Bessel Polynomials*, Lecture Notes in Math. 698, Springer, Berlin, 1978.
[16] P. C. Hansen, *Test matrices for regularization methods*, SIAM J. Sci. Comput., 16 (1995), pp. 506–512.
[17] E. Hille, *Introduction to general theory of reproducing kernels*, Rocky Mountain J. Math., 2 (1972), pp. 321–368.
[18] C. V. M. van der Mee, M. Z. Nashed, and S. Seatzu, *Sampling expansions and interpolation in unitarily translation invariant reproducing kernel Hilbert spaces*, Adv. Comput. Math., 19 (2003), pp. 355–372.
[19] C. V. M. van der Mee, G. Rodriguez, and S. Seatzu, *Semi-infinite multi-index perturbed block Toeplitz systems*, Linear Algebra Appl., 366 (2003), pp. 459–482.
[20] A. Ron and Z. Shen, *Frames and stable bases for shift invariant subspaces of* $L_2(\boldsymbol{R}^d)$, Canad. J. Math., 47 (1995), pp. 1051–1094.
[21] R. Schaback, *Error estimates and condition numbers for radial basis function interpolation*, Adv. Comput. Math., 3 (1995), pp. 251–264.
[22] G. Strang, *A proposal for Toeplitz matrix calculations*, Stud. Appl. Math., 74 (1986), pp. 171–176.
[23] M. Tismenetsky, *A decomposition of Toeplitz matrices and optimal circulant preconditioning*, Linear Algebra Appl., 154/156 (1991), pp. 105–121.
[24] E. E. Tyrtyshnikov, *Optimal and superoptimal circulant preconditioners*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 459–473.
[25] E. T. Whittaker and G. N. Watson, *A Course of Modern Analysis*, 4th ed., Cambridge University Press, London, 1927.
[26] R. M. Young, *An Introduction to Nonharmonic Fourier Series*, 2nd ed., Academic Press, New York, 2000.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 4, pp. 1150–1178

AN ITERATIVE METHOD FOR SOLVING COMPLEX-SYMMETRIC SYSTEMS ARISING IN ELECTRICAL POWER MODELING*

VICTORIA E. HOWLE† AND STEPHEN A. VAVASIS‡

Abstract. We propose an iterative method for solving a complex-symmetric linear system arising in electric power networks. Our method extends Gremban, Miller, and Zagha's [in *Proceedings of the International Parallel Processing Symposium*, IEEE Computer Society, Los Alamitos, CA, 1995] support-tree preconditioner to handle complex weights and vastly different admittances. Our underlying iteration is a modification to transpose-free QMR [6] to enhance accuracy. Computational results are described.

Key words. iterative methods, preconditioning, complex-symmetric systems, support-tree preconditioning, electrical power networks

AMS subject classifications. 65F10, 65F50

DOI. 10.1137/S0895479800370871

1. AC power networks.

Consider the linear system

$$A^T D^{-1} A\mathbf{v} = A^T D^{-1}\mathbf{b} \tag{1}$$

in which A is an $m \times n$ real matrix, D is an $m \times m$ complex diagonal matrix whose diagonal entries have positive real parts, $\mathbf{b}$ is a complex m-vector, and $\mathbf{v}$ is the n-vector of unknowns.

Equation (1) arises in the analysis of an alternating-current (AC) electrical network composed of generators and loads joined by a graph. Each node in the graph has a voltage, which is a complex number. The magnitude of the complex number is the magnitude of the voltage, and the argument is the phase difference of the voltage with respect to some reference phase.

Similarly, currents in the system are also complex numbers associated with graph edges. The generators can be modeled as voltage sources with a fixed voltage. The loads can be modeled as devices with fixed impedance. The impedance is a complex number with a positive real part.

If one is given the voltages of the generators and the impedances of the loads, then the problem of recovering the voltages at all nodes reduces to solving linear equations of the form (1). In this case, A is the *node-arc incidence matrix* (NAI) of the network. An NAI of a directed graph has one row for every edge of the graph and one column for every node. In each row, all entries are zeros except for exactly one "1" and one "−1" per row, which correspond to the endpoints of the graph edges. The diagonal

*Received by the editors April 10, 2000; accepted for publication (in revised form) by R. Freund July 26, 2004; published electronically May 6, 2005. This work was supported in part by NSF grant CCR-9619489, NSF grant DMS-9505155, and ONR grant N00014-96-1-0050. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.
http://www.siam.org/journals/simax/26-4/37087.html
†Sandia National Laboratories, P.O. Box 969, MS 9159, Livermore, CA 94551 (vehowle@sandia.gov).
‡Department of Computer Science, Cornell University, Ithaca, New York 14853 (vavasis@cs.cornell.edu).

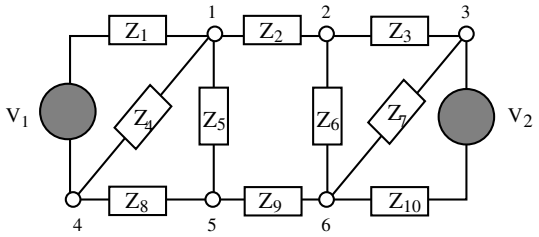


FIG. 1. *A simple AC network with two generators. Each edge j has a given constant impedance Z_j.*

matrix D stores the impedances of the loads, $\mathbf{b}$ holds the generator voltages, and $\mathbf{v}$ is the vector of node voltages. The linear system (1) is obtained from Ohm's law and Kirchhoff's law (current balance):

$$\begin{aligned} D\mathbf{i} + A\mathbf{v} &= \mathbf{b} \quad \text{(Ohm)}, \\ A^T\mathbf{i} &= \mathbf{0} \quad \text{(Kirchhoff)}. \end{aligned} \tag{2}$$

If we multiply Ohm's law by $A^T D^{-1}$ and apply current balance, we obtain the linear system (1).

As an example of the various components in (1), consider the simple network in Figure 1.

For this network,

$$A = \begin{bmatrix} 1 & & & -1 & & \\ 1 & -1 & & & & \\ & 1 & -1 & & & \\ 1 & & & -1 & & \\ 1 & & & & -1 & \\ & 1 & & & & -1 \\ & & 1 & & & -1 \\ & & & 1 & -1 & \\ & & & & 1 & -1 \\ & & 1 & & & -1 \end{bmatrix},$$

$$D = \operatorname{diag}(\begin{bmatrix} Z_1 & Z_2 & Z_3 & Z_4 & Z_5 & Z_6 & Z_7 & Z_8 & Z_9 & Z_{10} \end{bmatrix}),$$

$$\mathbf{b} = \begin{bmatrix} V_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & V_2 \end{bmatrix}^T. \tag{3}$$

When there are no faults, the diagonal elements of D, i.e., the impedances, are of approximately the same magnitude. When there is a fault in the network, e.g., a nearly open circuit exists in transmission lines, some of the impedances are much larger than the impedances associated with the functioning edges, making D extremely ill conditioned. See Bergen [1] for more information about modeling AC networks.

In the network shown in Figure 2, there are nearly open circuits in the edges associated with impedances Z_2 and Z_9. For this system, the matrix A and the vector $\mathbf{b}$ would be the same as those for the system in Figure 1, but the matrix D would now contain impedances of greatly varying magnitudes, e.g.,

$$D = \operatorname{diag}(\begin{bmatrix} Z_1 & \boldsymbol{Z_2} & Z_3 & Z_4 & Z_5 & Z_6 & Z_7 & Z_8 & \boldsymbol{Z_9} & Z_{10} \end{bmatrix}), \tag{4}$$

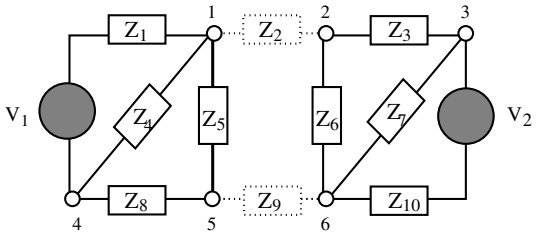


FIG. 2. *A simple AC network with two generators. Each edge j has a given constant impedance Z_j. In this example, we show faults in the edges associated with impedances Z_2 and Z_9. In the case of a nearly open circuit, for example, the magnitudes of Z_2 and Z_9 would be much greater than the magnitudes of the other impedances.*

where the $\boldsymbol{Z_2}$ and $\boldsymbol{Z_9}$ values are much larger in magnitude than the other impedances. Modeling faulty networks is important in practice since load-regulating devices must be designed to function properly even if part of the network fails.

We assume throughout this paper that the gap between the magnitudes of high impedance wires and low impedance wires is large and that removal of the high impedance wires disconnects the graph. In this case, the matrix $K = A^T D^{-1} A$ can be arbitrarily ill conditioned. It is not required that the removal of high impedance wires disconnect the graph for our algorithm to work; however, it is for this case that our method is a significant improvement over previous work [9].

We make two main contributions in this paper. The first contribution is an extension of Gremban's support tree preconditioner to cover complex weights (i.e., AC networks) and widely varying edge weights (i.e., faults). Even once we have a good preconditioner M, in the presence of a fault that disconnects the graph, K and therefore M^{-1} can be extremely ill conditioned separately. Even though the product $M^{-1}K$ is well conditioned, $M^{-1}(K\mathbf{v})$ may be computed inaccurately. Our second contribution is a technique that computes $M^{-1}(K\mathbf{v})$ accurately by splitting $K\mathbf{v}$ into its components in the range and null space of the functioning edges. For our algorithm to work efficiently, we need an efficient projection into the range and null spaces of the functioning edges.

A more general approach to achieving high accuracy in this kind of layered system was proposed by Bobrovnikova and Vavasis [3]. The method of [3] does not assume that there is an efficient projection into the range and null space of the functioning edges. But that method appears to be very difficult to precondition.

A direct algorithm known as complete orthogonal decomposition was proposed by Hough and Vavasis [13]. This method applies to the weighted least squares problem associated with faulted DC power networks. However, the method relies on the system being real and positive definite. There is no simple extension to the complex-symmetric case.

Other previous related work includes another combinatorial preconditioner for weighted node-arc adjacency matrices by Guo and Skeel [11], previous versions of support-tree preconditioners by Vaidya [18] and Bern et al. [2], and work by Vuik, Segal, and Meijerink [20] on a related mathematical problem arising in diffusion modeling using an explicit eigenvector projection. The problem analyzed by Vuik, Segal, and Meijerink involves a real, symmetric, positive definite matrix that is highly ill conditioned due to a large contrast in permeability coefficients in the system being modeled. The method proposed by Vuik, Segal, and Meijerink relies on a good choice

of projection vector, which involves knowing properties of the eigenvectors.

From the electrical power modeling perspective, there has been some work on using iterative methods for solving the complex-symmetric systems arising in electrical power modeling; see, e.g., [4], [7], [15], [17]. However, these methods have not addressed the ill-conditioning associated with faults in the electrical power system.

2. Support trees. Our system (1) is singular because the nodes are ungrounded. Since the voltage values are potentials, if we have not set one of the nodes to some reference voltage (i.e., grounded the node), we can add an arbitrary constant to all of the voltages and have an equally valid solution. Mathematically, this means that the matrix $A^T D^{-1} A$ is singular, because the vector of all 1's is in the nullspace of A. We address this detail by projecting vectors onto the range space of $A^T D^{-1} A$. This projection is ignored for the remainder of the paper. After grounding the system, it is still ill conditioned for two reasons.

The first source of ill-conditioning is inherent in NAI matrices. For example, if A describes an $n \times n$ grid-graph, $\kappa(A^T A) = O(n^2)$. This ill-conditioning has been addressed by *support-tree* preconditioners developed by Gremban, Miller, and Zagha [10]. We also use support-tree preconditioners in our method. The second source of ill-conditioning is caused by the widely varying weights in the faulted system. Gremban, Miller, and Zagha analyze only the case of nearly equal weights. We extend their analysis of condition numbers of the preconditioned system first to subgraphs having edge weights with widely varying magnitudes and then to graphs with complex edge weights.

2.1. Support trees of Gremban, Miller, and Zagha. Gremban, Miller, and Zagha form a support-tree preconditioner as follows. First, divide the nodes of the network graph into some number of approximately equal-sized subgraphs. Then recursively subdivide the subgraphs, etc., until all of the individual nodes have been separated. Note that the method does not depend on the number of subgraphs in each subdivision. For the results in this paper, we recursively subdivide into quarters. Next, build a tree based on this partitioning. The root of the tree is in correspondence with the entire original graph. The children of the root are in correspondence with the subgraphs of the graph obtained from the first partition, and so on down to the leaves of the tree, which are in correspondence with the individual nodes of the original graph. (See Figure 3.)

Next assign weights to the edges of this tree based on the edge weights in the original graph. Let G be the original network graph and let S be the support tree. Let v be a support-tree node corresponding to subgraph V of G, and let e be the support-tree edge from v to its parent. Assign weight to e equal to the sum of the conductances (i.e., reciprocal resistances) of edges in G connecting V to $G - V$. In other words, the weight on e is the sum of the entries of D^{-1} corresponding to the nodes of V.

DEFINITION 1. *We define the* weighted Laplacian matrix $L(G)$ *of an* n*-node graph* G *as the* $n \times n$ *matrix whose* j*th diagonal entry corresponds to the sum of weights of the edges incident on the* j*th node of the graph. The* (i, j) *entry of* $L(G)$ *is equal to the negative of the weight of the edge in* G *connecting nodes* i *and* j*.*

Note that the system matrix K is the weighted Laplacian of the input graph G. Let T be the weighted Laplacian matrix of the new network S. If n is the number of original circuit nodes and t is number of nonsingleton subgraphs created during partitioning, then $n + t$ is the number of support-tree nodes. The matrix T is an $(n + t) \times (n + t)$ very sparse matrix.

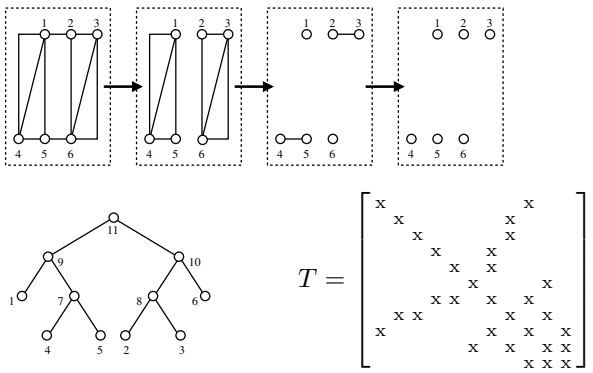


FIG. 3. *Example of constructing a support tree. The top left picture shows a network graph with six nodes and ten edges. Successive cuts into subgraphs are shown in the pictures progressing left to right. The support tree is shown on the bottom left, with the corresponding Laplacian matrix T on the bottom right. Notice that the leaves of the support tree correspond to the nodes of the network graph, and the root of the support tree corresponds to the entire original network graph.*

Let M be the Schur complement of T obtained by eliminating the internal tree nodes, that is, the tree nodes that are not leaves. Gremban, Miller, and Zagha showed that M is a good preconditioner for $A^T D^{-1} A$ in the real case (DC) with uniform edge weights (no faults). In particular, they showed that for equally weighted instances of (1) (i.e., $D = I$), the condition number of grid-graphs is reduced from $O(n^2)$ to $O(n \log n)$. Although M is dense, linear systems of the form $M\mathbf{v} = \mathbf{r}$ can nonetheless be solved in linear time using Cholesky factorization on the larger sparse matrix T. Note that T has a perfect elimination order since it is the weighted Laplacian matrix of a tree and we can eliminate from the leaves to the root. They show that solving

$$\begin{pmatrix} T_1 & T_2 \\ T_2^T & T_3 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ \mathbf{0} \end{pmatrix} \tag{5}$$

and letting $\mathbf{v} = \mathbf{v}_1$ are equivalent to solving $M\mathbf{v} = \mathbf{r}$, where $M = T_1 - T_2 T_3^{-1} T_2^T$ is the support-tree preconditioner. Thus, the preconditioner is efficient in practice.

2.2. Extensions of the support-tree preconditioner. To extend the idea of support-tree preconditioners to the case of an AC network with faults, we change how we build the support tree as follows. First, assume that removal of the faulted edges disconnects the graph into at least two subgraphs. (If this assumption does not hold, then our method does not constitute an improvement over previous work [9].) We require the top level of the support tree to be composed entirely of faulted edges, and children of the root should be connected subgraphs of the network after faulty edges are deleted. We include all of the faulted edges in the first separator. This in turn puts all of the weights corresponding to faulted edges in the top level of the support tree. We build the rest of the support tree as before. We show below that with this change, the support-tree preconditioners are good preconditioners for $A^T D^{-1} A$. In applying the preconditioner M in the AC case, the matrices $K = A^T D^{-1} A$ and M are complex-symmetric and hence the Cholesky factorization technique does not apply directly. We show that the LU factorization can still be stably performed

without pivoting (and thus without fill-in). Thus, as before, we can solve systems of the form $M\mathbf{v} = \mathbf{r}$ by solving the larger sparse system involving T.

THEOREM 2. *T has an LU factorization, and the elements of L are bounded; i.e., the computation is stable without pivoting.*

Proof. The diagonal elements of T are exactly the negative sums of the off-diagonal elements. In addition, because we have ordered the tree nodes from the leaves to the root, T has perfect elimination order. The elements of L are thus only 0's, 1's, and -1's. The L matrix has 1's on the diagonal, -1's in the elements of T below the diagonal that are nonzero, and 0's elsewhere. ∎

In addition to the ill-conditioning inherent in NAI matrices, there is also ill-conditioning in our application of (1) caused by widely varying weights in a faulted system. Let ρ_1 be the absolute value of a typical admittance in functioning wires, and let ρ_2 be the absolute value of a typical admittance in faulty wires. Then $\rho_1 \gg \rho_2$, and the original linear system can be decomposed as $K = A^T D^{-1} A = K_1 + K_2$, where $K_1 = A_1^T D_1^{-1} A_1$ and $K_2 = A_2^T D_2^{-1} A_2$. Here subscripts $1, 2$ denote the partition into functioning and faulty wires, respectively; hence $\|D_1\| \approx \rho_1$ and $\|D_2\| \approx \rho_2$.

Gremban, Miller, and Zagha analyze only the case of nearly equal weights. We extend their analysis of condition numbers of the preconditioned system first to graphs having edge weights with widely varying magnitudes, assuming the change in forming the support tree that we discussed above and then to graphs with complex edge weights. We assume for simplicity in the following theorems that the faults are nearly open circuits, i.e., low admittance wires.

2.3. Extensions to analysis. In the rest of this section, we extend Gremban's analysis as follows. Since electrical power networks are laid out geographically, $n \times n$ grid-graphs are a reasonable first approximation to consider. Therefore, we first extend Gremban's analysis to DC networks (i.e., networks in which the edges have real weights) with faults by proving that in the DC grid-graph case with faults along the median edges, the condition number of the preconditioned system is $O(n \log n)$, where n^2 is the number of nodes in the system. The median edges are the edges running through the middle of the grid-graph, horizontally and vertically. That is, removal of the median edges would divide the grid-graph into four approximately equal-sized subgraphs. Since in general electrical power networks are not actually laid out on grid-graphs and faults are not confined to being along the median edges, we show results for general graphs. Little is known about the behavior of support-tree preconditioners on general graphs. Although we do not have an upper bound on the condition number of the preconditioned system on a general graph, we can show that for general DC networks with arbitrarily located faults, there is a bound on the condition number of the preconditioned system that is independent of the relative magnitude of the faults. Finally, we extend our results for grid-graphs and general graphs to the AC case. Given certain assumptions about the impedance values in the network, we show that we can bound the condition number in the AC faulted case based on the condition number of the related DC network obtained by taking the real part of the impedance values. In particular, we assume that the impedance values in the network (the diagonal elements of D) lie in a pointed cone in the complex plane; i.e., if d_j is a diagonal element of D, then $d_j = x_j + iy_j$ where $x_j > 0$ and $|y_j| \leq \mu x_j$ for some positive cone constant μ. This series of theorems then shows that we can extend Gremban's support-tree preconditioner, with certain changes, to be a good preconditioner in the case of AC networks with faults.

2.3.1. Extensions to analysis of DC grid-graph networks with faults. We first extend Gremban's analysis to the case of DC (real) networks with faults. We first introduce some notation that will be useful in the proofs that follow. Note that the definitions that do not specifically rely on the matrices being real apply equally to the AC case.

DEFINITION 3. *We refer to the* condition number *of the preconditioned system* $M^{-1}K$ *as* $\kappa(M,K)$*, where*

$$\kappa(M,K) = \max \frac{|\mathbf{x}^* M \mathbf{x}|}{|\mathbf{x}^* K \mathbf{x}|} \cdot \max \frac{|\mathbf{x}^* K \mathbf{x}|}{|\mathbf{x}^* M \mathbf{x}|}, \tag{6}$$

where the maxima are taken over nonzero vectors $\mathbf{x}$ *in the range space of* K *(which is equal to the range space of* M*). In the real case, the absolute value symbols are not needed.*

Note that this definition is equivalent to the usual eigenvalue definition of condition number in the real case.

We next prove a series of lemmas that we will use to show that in the DC (real) case, the upper bound on the condition number of the preconditioned system does not depend on the values of the faulted edges, and that for $n \times n$ (real) grid-graphs, the condition number is $O(n \log n)$. The key technique we will use is support numbers.

DEFINITION 4 (see [9]). *For two real positive semidefinite matrices* A *and* B*, the support of* B *for* A*,* $\sigma(A,B)$*, is defined to be the greatest lower bound over all* τ *such that* $\tau B - A$ *is positive semidefinite.*

Gremban relates this quantity to the condition number as follows.

LEMMA 5. *Let* A, B *be real positive semidefinite matrices. Then* $\kappa(A,B) = \sigma(A,B)\sigma(B,A)$.

Thus, for our grid-graph construction, we must obtain upper bounds on $\sigma(M,K)$ and $\sigma(K,M)$.

We start with $\sigma(M,K)$. The support tree for the grid-graph is not completely regular because subgraphs that are adjacent to the boundary of the entire grid have fewer edges emanating from them than subgraphs on the interior. Our analysis of $\sigma(M,K)$ is simplified by assuming, however, that all subgrids of size $2^h \times 2^h$ have exactly $4h$ edges emanating from them. This assumption may be made without loss of generality for the following reason. Let T' be the network resulting from augmentation of T with these extra edges. Then $T' - T$ is positive semidefinite (since inserting edges corresponds to adding a semidefinite matrix); hence so is $M' - M$, where M' is the Schur complement of T' according to the following theorem.

THEOREM 6 (see [12]). *Let* A, B *be* $n \times n$ *symmetric positive semidefinite matrices of the same size, and let* k *be an integer between* 1 *and* $n-1$*. Assume that the upper left* $k \times k$ *submatrices of both* A *and* B *are invertible, and let* $\mathrm{Schur}_k(A)$ *denote the Schur complement of the upper left* $k \times k$ *submatrix (i.e.,* $\mathrm{Schur}_k(A) = A(k+1:n, k+1:n) - A(k+1:n, 1:k)A(1:k, 1:k)^{-1}A(1:k, k+1:n)$*). Similarly, let* $\mathrm{Schur}_k(B)$ *be the Schur complement of the upper left* $k \times k$ *submatrix of* B*. Then if* $A - B$ *is positive semidefinite, so is* $\mathrm{Schur}_k(A) - \mathrm{Schur}_k(B)$.

This means that if τ is a scalar such that $\tau K - M'$ is positive semidefinite, so is $\tau K - M' + (M' - M) = \tau K - M$. Therefore, $\sigma(M,K) \leq \sigma(M',K)$, so any upper bound for $\sigma(M',K)$ applies also to $\sigma(M,K)$.

Hence we assume T has the regular structure mentioned earlier for the remainder of this analysis. The next step in the analysis of $\sigma(M,K)$ is to obtain upper bounds for the off-diagonal entries of M. This is the purpose of the next three lemmas.

LEMMA 7. *Let M be the Schur complement of T as described earlier. Then $M(j,k)$ is the kth entry of $\mathbf{i}$, where $\mathbf{i}$ and $\mathbf{y}$ satisfy the following equation in which $\mathbf{e}_j$ denotes the jth column of the identity matrix:*

$$T \begin{bmatrix} \mathbf{y} \\ \mathbf{e}_j \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{i} \end{bmatrix}. \tag{7}$$

Remark 1. This lemma has an interpretation in terms of electrical networks. Recall that multiplying a weighted Laplacian matrix of a graph by a vector $\mathbf{v}$ corresponds to assigning voltages to the nodes given by $\mathbf{v}$ and then determining the excess currents at the nodes. Therefore, the above lemma corresponds to holding leaf node j of the support tree to voltage equal to 1, the other leaf nodes to voltage 0, letting the nonleaf tree nodes "float" (i.e., assume whatever voltage is needed so that the current at the node balances), and measuring the excess current at node k.

Proof. This lemma holds because M is obtained from T by performing Gaussian elimination steps on (7) to eliminate the nonleaf tree nodes. If we perform Gaussian elimination on (7), we obtain $M\mathbf{e}_j = \mathbf{i}$; i.e., the kth entry of $\mathbf{i}$ is equal to $M(j,k)$. □

There is exactly one path between nodes j and k in the support tree. In estimating $\mathbf{i}(k)$, we need to consider two cases: either the path between nodes j and k goes through the root node (Lemma 8), or it does not (Lemma 9).

LEMMA 8. *Let M be the support-tree preconditioner for the $n \times n$ grid-graph with edge weights described above. Assume n is an exact power of 2. Let l be the number of levels in the tree, i.e., $n = 2^l$, assuming exact quadrisection at each level. Let the median (faulted) edges have admittance $\epsilon/4$ and the nonfaulted edges have admittance $1/4$. Assume the path from j to k passes through the root node of the support tree. Then*

$$|M(j,k)| \leq \frac{49 \cdot 2^{-3l}\epsilon}{288} + O(2^{-6l}\epsilon) + O(\epsilon^2).$$

Remark 2. The assumption about powers of 2 is made to simplify the proof and the notation. The factor of $1/4$ unclutters the figures but is otherwise unnecessary.

Proof. Although this lemma can be proved using purely algebraic arguments, we prefer to argue using principles of electrical networks because we believe this gives more insight.

The first part of the proof contracts T almost to a path using series-parallel equivalent circuits. A similar argument was used by Gremban. For example, consider two leaf nodes of T, both holding a voltage 0, attached to the same parent node p with edges that both have resistance r. These two leaf nodes can be merged into a single node of voltage 0 connected to the same parent with resistance $r/2$. Then, since p is a floating node and is connected to exactly two edges with resistance r_1 and r_2, p can be deleted and the two edges can be replaced by a single edge with resistance $r_1 + r_2$. Proceeding in this manner, we can merge and contract many tree nodes as shown in Figure 4.

As in Gremban's analysis, if we reduce nodes up to a child w of a node u at level $(i-1)$, the reduced system is equivalent to a node connected to u with an edge resistance less than or equal to $1/2^{i-1}$. (It can be shown using electrical reasoning or algebra that overestimating the resistance of an off-path edge will lead to overestimation of $|M(j,k)|$, which is valid since we are trying to obtain an upper bound on this quantity.)

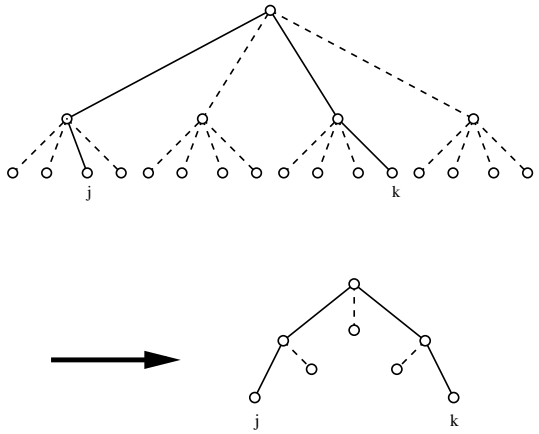


FIG. 4. *The solid lines in the first tree show the path between nodes j and k. The dashed lines are the edges that we reduce up to the (j,k)-path. The second tree shows the result of reducing all other nodes up to the path between j and k. The (j,k)-path is again shown with solid lines, with the reduced edges shown as dashed lines.*

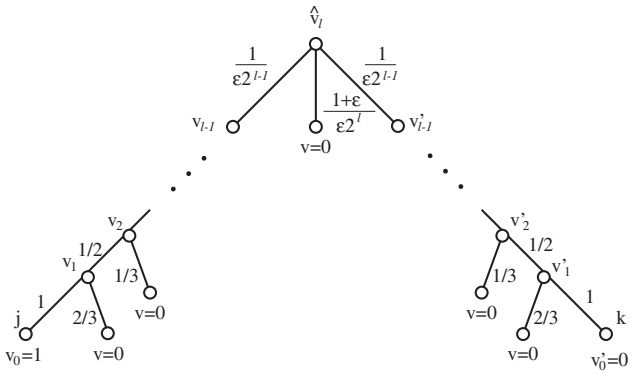


FIG. 5. *Support-tree reduced to the (j,k)-path. The node $\hat{v}'_l$ represents the root of the tree, v_0 represents the j-node, and v'_0 represents the k-node. The edge labels shown are resistances.*

Therefore, the reduced system will look like the system in Figure 5 for a power-of-two grid. In this figure, v_0 represents the jth node of the tree (whose voltage has been set to one), v'_0 represents the kth node of the tree (whose voltage has been set to zero), and $\hat{v}_l$ represents the root of the tree.

Once we have reduced the system to the (j,k)-path, we solve for the net current at the kth node, i.e., at v'_0 in Figure 5. Recall that finding this current is equivalent to finding $\mathbf{i}(k) = M(j,k)$.

Solving current balance equations at each floating node, we get the recurrence relations

$$4v_{i+2} - 9v_{i+1} + 2v_i = 0, \tag{8}$$
$$4v'_{i+2} - 9v'_{i+1} + 2v'_i = 0.$$

We also have equations for the root node and its neighbors. Using standard

techniques, we can solve these recurrences to obtain

$$\mathbf{i}_k = \frac{49\epsilon 2^{-3l}}{288} + O(\epsilon 2^{-6l}) + O(\epsilon^2), \tag{9}$$

which bounds the element $M(j,k)$ in the case where the (j,k)-path goes through the root of the support tree. For details of the analysis, see [14]. ∎

Next we deal with the case where the (j,k)-path does not pass through the root of the tree.

LEMMA 9. *Let M be the support-tree preconditioner as defined above. Then $|M(j,k)| \le 3.5/8^h$ assuming the (j,k)-path does not pass through the root of the tree. Here h is the height of the common ancestor of j,k in the support tree.*

Proof. As in Lemma 8, we begin by reducing up to the (j,k)-path. In this case, after reducing the tree as before to the (j,k)-path, we have a completely unfaulted path between nodes j and k in the support tree, with an extra path off of the root $\hat{v}_l$ containing all of the faulted edges and other edges hanging off the path. The same recursion (8) applies to this analysis, and the same techniques can be used to solve it. ∎

The approach used for estimating $\sigma(M,K)$ is the same as Gremban's; namely, we first partition the graph and the preconditioner, and then for each piece we apply a congestion/dilation argument. The first lemma concerns partitioning.

LEMMA 10. *If $A = A_1 + \cdots + A_s$ and $B = B_1 + \cdots + B_s$, where each A_i and each B_i is symmetric positive semidefinite, then*

$$\sigma(A,B) \le \max\{\sigma(A_1,B_1), \ldots, \sigma(A_s,B_s)\}.$$

Proof. This lemma is proved by Gremban [9, Chapter 4]. ∎

The particular partitioning to be used in our analysis is as follows. As in Gremban, we write $K = K_1 + \cdots + K_l$ and $M = M_1 + \cdots + M_l$. The partition of K is given by $K_h = 2^{h-l-1}K$. Thus, $K_l = K/2$, $K_{l-1} = K/4$, etc. (This leaves the fraction 2^{-l} of K "unused." This is valid because underestimating K can only increase $\sigma(M,K)$.) The partition of M is given by the rule that $M_h(i,j) = M(i,j)$ provided that the highest tree node in the support tree on the (i,j)-path is at height h; else $M_h(i,j) = 0$. (We order "height" so that the leaves are at height 0 and the root at height l.)

The next part of the analysis of $\sigma(M,K)$ involves a congestion-dilation argument. The terms are defined as follows.

DEFINITION 11 (see [9]). *Let A and B be graphs. Let p_0 be an injective mapping from nodes of A into nodes of B. Suppose there exists a mapping p from edges in A to paths in B such that if $e = (a,b)$ is an edge in A,then the endpoints of $p(e)$ are $(p_0(a), p_0(b))$. For an edge f in B, let $d_1, \ldots, d_k$ be the edges in A such that $f \in p(d_i)$; that is, d_i are the A edges whose embedding path in B includes edge f. The* congestion *of edge f is*

$$\gamma(A,B,f) = \frac{\sum_{i=1}^{k} weight(d_i)}{weight(f)}. \tag{10}$$

The congestion *of the mapping $\gamma(A,B)$ is the maximum congestion over all edges in B.*

DEFINITION 12 (see [9]). *Given an embedding of a graph A into a graph B as in Definition* 11*, the* dilation *of the embedding is defined to be the length of the longest path in B onto which an edge of A is mapped.*

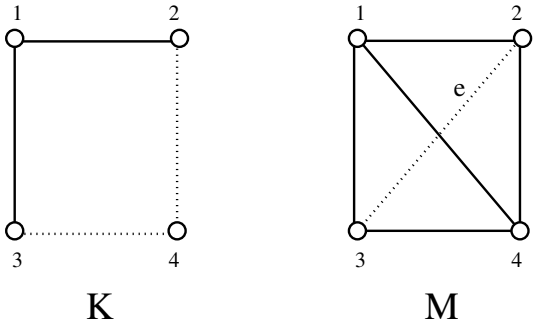


FIG. 6. *The graph K on the left is the original four-node system, and the graph M on the right is the graph corresponding to the Schur complement support-tree preconditioner M. Consider edge $e \in M$ (the dotted edge in the figure connecting M vertices $v_M = 2$ and $w_M = 3$). The corresponding path in K is the one shown by dotted edges in K (connecting K-nodes $v_K = 2$ to 4 to $w_K = 3$).*

LEMMA 13. *The support $\sigma(A, B)$ of a matrix B for a matrix A is less than or equal to the maximum congestion over edges in B times the maximum dilation over paths in B.*

Proof. This lemma is proved by Gremban [9, Chapter 4]. □

In order to apply the congestion–dilation analysis, we assume the following standard mapping of the edges of M onto paths in the grid-graph corresponding to K. Let e be an edge in M. The nodes of M correspond to the nodes in the original network K. If edge $e \in M$ connects nodes v_M and w_M in M, we map edge e to the path in K obtained by starting at the node $v_K \in K$ that corresponds to v_M and moving vertically until we reach the same row as node w_K (the node in K corresponding to node w_M). We then complete the path by moving horizontally until we reach node w_K.

As an example, consider the very simple four-node system shown in Figure 6. Form the support tree by bisecting the graph repeatedly, form the weighted Laplacian matrix T from the support tree, and then let M be the Schur complement obtained by performing Gaussian elimination on the internal (nonleaf) nodes of the tree. The graph corresponding to the preconditioner M is then as shown in the same figure. Consider edge $e \in M$ (the dotted edge in Figure 6 connecting M vertices $v_M = 2$ and $w_M = 3$). The corresponding path in K is the one shown by dotted edges in K (connecting K-nodes $v_K = 2$ to 4 to $w_K = 3$).

Now finally we have enough tools to estimate $\sigma(M, K)$.

LEMMA 14. *Let $K = A^T D^{-1} A$ correspond to a DC (real weights) network on an $n \times n$ grid-graph containing wires of two magnitudes, functioning wires with conductance $1/4$ and faulty wires with conductance $\epsilon/4$, where the faulty wires are along the median edges of the grid. Let M be the support-tree preconditioner as defined above. Then $\sigma(M, K) = O(n)$.*

Proof. As explained above, we estimate $\sigma(M_h, K_h)$ for each $h = 1, \ldots, l$ by considering $\gamma(M_h, K_h) \cdot \delta(M_h, K_h)$. Let us pick an h and select an e in K. We try to determine $\gamma(M_h, K_h, e)$. There are two cases to consider: either e is one of the median edges of K whose conductivity is $\epsilon/4$ or it is another edge.

Let us consider the median-edge case first. We observe that $\gamma(M_h, K_h, e) = 0$ if $h < l$ since no edge in M_h corresponds to a path that passes through the root node. This is because for $h < l$, all the edges in M_h correspond to K-paths that lie inside one of the four quadrants of K (after the median edges are removed). Thus, e does

not support any of these M-edges.

Thus, for a median edge we need consider only $\gamma(M_l, K_l, e)$. Let $d_1, \ldots, d_p$ be the edges in M_l whose corresponding K-paths $s_1, \ldots, s_p$ pass through e. Note that the tree paths corresponding to $d_1, \ldots, d_p$ must all pass through the root of the support tree. Therefore, the weight in M_l of each of these p edges is $(49\epsilon 2^{-3l})/288 + O(\epsilon 2^{-6l}) + O(\epsilon^2)$ by Lemma 8. Next, we have $p \le 2^{3l}$. This is because e is either in the same row (if e is horizontal) or column (if e is vertical) of one of the two endpoints of each s_i. Therefore, the number of possible endpoints for one end of s_i is n; the number of possible endpoints for the other end is n^2 (the total number of nodes). Thus, $p \le n^3 = 2^{3l}$. The weight in K_l of this edge e is $\epsilon/8$ (since $K_l = K/2$). Therefore, the maximum congestion in this case is

$$\begin{aligned}\gamma(M_l, K_l, e) &= \frac{\sum_{\mu=1}^{p} \frac{49\epsilon 2^{-3l}}{288} + O(\epsilon 2^{-6l}) + O(\epsilon^2)}{\text{weight}(e)} \\ &= \frac{O(2^{3l}\frac{49}{288}\epsilon 2^{-3l})}{\epsilon/8} \\ &= O(1).\end{aligned}$$

Next, consider the case that e is a functioning edge of K. Let (i, j) be the endpoints of e. Let h' be the height of the highest node z on the support-tree path connecting i to j. By assumption for this case, $h' < l$. If $h < h'$, then $\gamma(M_h, K_h, e) = 0$ because no paths induced by M_h-edges can use edge e, since any path in K induced by an edge of M_h uses nodes whose highest common ancestor in the support tree is at level h.

Thus, assume $h \ge h'$. Let $d_1, \ldots, d_p$ be the edges in M_h whose corresponding K-paths $s_1, \ldots, s_p$ pass through e. This means that $s_1, \ldots, s_p$ are all contained in a $2^h \times 2^h$ subgrid that also contains e, namely, the subgraph of K corresponding to the nodes in the support tree at z', where z' is the unique ancestor of z at height h. The total number p of such paths that can use e is therefore at most 8^h. This is because, depending on whether e is vertical or horizontal, one endpoint of each s_i is either in the same row or in the same column as e, so there are at most 2^h choices for one of the endpoints. The other endpoint could be anywhere in the subgrid determined by z', which has 2^{2h} nodes total.

Note that the weight of any M_h-edge is at most $3.5/8^h$ by Lemma 9. The weight of e in K_h is $(1/4) \cdot 2^{h-l-1} = 2^{h-l-3}$ by definition of K_h and the assumption that e is a functioning edge. Thus,

$$\begin{aligned}\gamma(M_h, K_h, e) &= \frac{\sum_{\mu=1}^{p} \text{weight}(d_\mu)}{\text{weight}(e)} \\ &= \frac{8^h \cdot 3.5/8^h}{2^{h-l-3}} \\ &= O(2^{l-h}).\end{aligned}$$

Thus, we have shown in all cases (both functioning and faulty wires) that for all e, $\gamma(M_h, K_h, e) = O(2^{l-h})$; hence $\gamma(M_h, K_h) = O(2^{l-h})$. Next, we observe that $\delta(M_h, K_h) = 2^h$. This is because the path in K_h induced by a node in M_h lies in the subgraph induced by a node in the support tree at height h, a grid whose diameter is 2^h. Thus, by Lemma 13, $\sigma(M_h, K_h) \le O(2^{l-h}) \cdot 2^h = O(2^l) = O(n)$. This holds for all h, so by Lemma 10, $\sigma(M, K) = O(n)$. ∎

Next, we turn our attention to $\sigma(K, M)$. The following lemma greatly simplifies this analysis.

LEMMA 15. *Let K, M be two $n \times n$ symmetric positive semidefinite matrices. Let M be the Schur complement of the lower right block of a larger symmetric positive semidefinite matrix T as in* (5)*. Let $\tilde{K}$ be K extended with zeros so that it is the same size as T; i.e.,*

$$\tilde{K} = \begin{bmatrix} K & 0 \\ 0 & 0 \end{bmatrix}. \tag{11}$$

Then $\sigma(K, M) \leq \sigma(\tilde{K}, T)$.

Proof. See Gremban [9, Chapter 4] for the proof. This also follows directly from Theorem 6 as follows. Define $\tilde{K}_\epsilon = [K, 0; 0, \epsilon I]$, where $\epsilon > 0$ and I is the identity matrix. Then clearly K is the Schur complement of the lower right block of $\tilde{K}_\epsilon$, so $\sigma(K, M) \leq \sigma(\tilde{K}_\epsilon, T)$ by the theorem. Then take the limit as $\epsilon \to 0$. ∎

The analysis of $\sigma(K, M)$ is now fairly straightforward.

LEMMA 16. *Let $K = A^T D^{-1} A$ correspond to a DC (real weights) network on an $n \times n$ grid-graph containing wires of two magnitudes—high admittance (functioning) wires and low admittance (nearly open circuit) wires—where the faults are along the median edges of the grid. Let T be as defined above and let $\tilde{K}$ be as defined in Lemma* 15*. Then $\sigma(\tilde{K}, T) = O(\log n)$.*

Proof. In this case, we are considering how well the support-tree matrix T supports the extended system matrix $\tilde{K}$. The only edges of $\tilde{K}$ are the edges of K; therefore we need only to map edges of K onto paths of T. We assume the following standard mapping of edges of K onto paths in T. Let e be an edge in K connecting nodes v_K and w_K. The leaves of the support tree correspond exactly to the nodes of the original network. Since v_K and w_K are nodes in the original network, they correspond to two leaf nodes in the support tree. We map edge e to the unique path between the corresponding leaves v_T and w_T in the support tree.

As before, the support of matrix T for $\tilde{K}$ is less than or equal to the product of the maximum congestion over edges in T and the maximum dilation over paths in T; i.e., $\sigma(\tilde{K}, T) \leq \gamma(\tilde{K}, T)\delta(\tilde{K}, T)$. We bound these two quantities separately.

The maximum dilation comes from the edge in K that must be mapped through a path that goes through the root of T. The length of this path is $2 \log_2 n + 2$.

Next we find the maximum congestion. Let p be the mapping described above from edges in K to paths in T, let e be an edge in T, and let $d_1, \ldots, d_k$ be the edges in K such that $e \in p(d_i)$ for some i.

Assume that edge $e \in T$ connects nodes v_j and v_k, where v_j is a child node of v_k. Then we have defined the edge weights of T such that the weight of edge e equals the sum of the edges in K that connect the nodes associated with the leaves of the tree rooted at v_j to the rest of the graph. This sum is exactly $\sum_{i=1}^{k} \text{weight}(d_i)$. Therefore, the congestion over any edge in T is one.

Thus, the support $\sigma(\tilde{K}, T)$ is less than or equal to $1 \times O(\log n) = O(\log n)$. ∎

Using Lemma 5 and Lemmas 13 through 16, we can show that in the case of an $n \times n$ grid-graph with faults along the median edges of the grid, the condition number of the preconditioned system is $O(n \log n)$. We summarize this result in the following theorem.

THEOREM 17. *Let $K = A^T D^{-1} A$ correspond to a DC (real weights) network on an $n \times n$ grid-graph containing wires of two magnitudes—high admittance (functioning) wires and low admittance (nearly open circuit) wires—where the faults are along the*

median edges of the grid. Let the preconditioner M be as above. Then $\kappa(M,K) = O(n \log n)$.

Proof. This follows from Lemmas 5, 14, and 16. ∎

2.3.2. Extensions to analysis of general DC networks with faults. In the nongrid case, we show that the condition number of the preconditioned system has a bound that is independent of the magnitude of the fault.

Since the condition number of the preconditioned system is bounded above by the product of the support of K for M, $\sigma(M,K)$, and the support of T for $\tilde{K}$, $\sigma(\tilde{K},T)$, we prove this result by showing that $\sigma(M,K)$ and $\sigma(\tilde{K},T)$ are each independent of the magnitude of the fault. We do this by considering the congestion and dilation of the respective embeddings. We show that both the congestion and dilation are independent of ρ_1 and ρ_2 for the case of K supporting M and for the case of T supporting $\tilde{K}$.

LEMMA 18. *Let $K = A^T D^{-1} A$ correspond to a connected DC (real weights) network containing wires of two magnitudes—high admittance (functioning) wires and low admittance (nearly open circuit) wires. Let the preconditioner M be as above. Assume that the admittance of a typical functioning edge is ρ_1, and the admittance of a typical faulted edge is ρ_2. Then $\sigma(M,K)$ has an upper bound that is independent of ρ_1 and ρ_2.*

Proof. In this lemma, we are considering the case of K supporting M, $\sigma(M,K)$. Fix a mapping from edges in M to paths in K. Such a mapping exists by the assumption that the network is connected. We further require that if there exists a path from a to b in K using unfaulted edges, then such a path must be selected for the embedding. The dilation is given by the longest path in K onto which an edge of M is embedded. For any mapping we choose, this path can be as long as the longest path in the K graph, but this length does not depend on ρ_1 or ρ_2.

For the congestion, we follow a proof similar to that for the grid-graph case. As in that case, we need an upper bound on the elements of M, which will give us a bound on the edge weights of the graph associated with M. Choose an edge $e \in K$. There are two cases to consider: either e is a faulted edge or it is not a faulted edge. If e is not a faulted edge, then weight$(e) = \rho_1$. Let $d_1, \ldots, d_k$ be the M edges whose K-paths include edge e. We claim that these edges all have weight of approximately ρ_1. We show this by using the same process of branch/path reduction as in Lemma 14, reducing the support tree to the (j,k)-path that connects the leaf nodes of T that correspond to the nodes in K connected by edge e. We can get an upper bound on the current at node k by removing the faulted branch of the reduced system. We can do this because the extra edges associated with the faulted parts of the system will only make the current that gets to the kth node less. The current at the kth node is therefore $O(\rho_1)$ giving us that $M(j,k) = O(\rho_1)$. As before, the current at node k equals the value of $M(j,k)$, so the M edges whose K-paths include edge e have weights of ρ_1. Thus the congestion in this case is approximately $O(\rho_1)/\rho_1 = O(1)$ and is independent of ρ_1 and ρ_2.

If e is a faulted edge, then weight$(e) = \rho_2$. We again let $d_1, \ldots, d_k$ be the M-edges whose K-paths include edge e. We claim that in this case these edges have weights approximately equal to ρ_2. We again show this by using the same process of branch/path reduction as in Lemma 14. By construction, any K-path using e must be a path between different K_1-subgraphs of K. Therefore, the (j,k)-path in T must go through the root of the tree. Contracting as before to the (j,k)-path, we can see that the current at node k must be $O(\rho_2)$. Therefore, $M(j,k) = O(\rho_2)$, and the M

edges whose K-paths include edge e have weights of $O(\rho_2)$. Thus the congestion in this case is approximately $O(\rho_2)/\rho_2 = O(1)$ and is independent of ρ_1 and ρ_2.

Since we have now shown that the dilation and congestion are separately bounded independently of ρ_1 and ρ_2, their product, which is an upper bound on the support of K for M, $\sigma(K, M)$, is also bounded independently of ρ_1 and ρ_2. ∎

LEMMA 19. *Let $K = A^T D^{-1} A$ correspond to a connected DC (real weights) network containing wires of two magnitudes—high admittance (functioning) wires and low admittance (nearly open circuit) wires. Let the preconditioner M be as above. Assume that the admittance of a typical functioning edge is ρ_1, and the admittance of a typical faulted edge is ρ_2. Then $\sigma(\tilde{K}, T)$ has an upper bound that is independent of ρ_1 and ρ_2.*

Proof. In this lemma, we consider the support of T for $\tilde{K}$, $\sigma(\tilde{K}, T)$. We map the edges of K (which are the same as the edges of $\tilde{K}$) onto paths in T using the same mapping as in Lemma 14. As in the previous case, the dilation is given by the longest path in T onto which an edge of K is mapped. This path can be as long as the longest path in T but is nevertheless independent of ρ_1 and ρ_2. For the congestion, let e be an edge in T and let $d_1, \ldots, d_k$ be the edges in K whose embedding path in T includes edge e. If e is a faulted edge, that is, $\text{weight}(e) \approx \rho_2$, then the d_i are faulted edges of K since only faulted edges of K would be mapped to paths that pass through the faulted edges of T. Therefore, $\sum_{i=1}^{k} \text{weight}(d_i) \approx k\rho_2$. Since the weight of e is also approximately ρ_2 and the congestion is defined as the previous sum divided by the weight of e, the congestion is independent of ρ_1 and ρ_2. If e is not a faulted edge, then some of the d_i may be faulted edges and some may not. In this case, $\sum_{i=1}^{k} \text{weight}(d_i) = O(\rho_1 + \rho_2) = O(\rho_1)$. Since e is not a faulted edge, its weight is approximately ρ_1, and the congestion is again independent of ρ_1 and ρ_2. Since the congestion and dilation separately are independent of ρ_1 and ρ_2, so is their product, which is an upper bound on the support of T for $\tilde{K}$, $\sigma(\tilde{K}, T)$. ∎

THEOREM 20. *Let $K = A^T D^{-1} A$ correspond to a connected DC (real weights) network containing wires of two magnitudes—high admittance (functioning) wires and low admittance (nearly open circuit) wires. Let the preconditioner M be as above. Assume that the admittance of a typical functioning edge is ρ_1, and the admittance of a typical faulted edge is ρ_2. Then $\kappa(M, K)$ has an upper bound that is independent of ρ_1 and ρ_2.*

Proof. From Lemma 15, we have $\kappa(M, K) \leq \sigma(\tilde{K}, T)\sigma(M, K)$, where K, $\tilde{K}$, T, and M are as before. We have shown in Lemmas 18 and 19 that the upper bounds on $\sigma(K, M)$ and $\sigma(\tilde{K}, T)$ are each independent of ρ_1 and ρ_2. Therefore, their product, which is an upper bound on the condition number of the preconditioned system, $\kappa(M, K)$, is also independent of ρ_1 and ρ_2. ∎

2.3.3. Extensions to analysis of AC networks. Finally, we extend the analysis to the AC network case.

DEFINITION 21. *Let B be a matrix of the form $B = A^T D^{-1} A$, where A is real with full column rank and D is a diagonal matrix. We say that B is* cone positive definite *if the diagonal elements of D lie in a pointed cone in the complex plane. In other words, if d_j is a diagonal element of D, then $d_j = x_j + iy_j$, where $x_j > 0$ and $|y_j| \leq \mu x_j$ for some positive cone constant μ.*

We assume that the original system $K = A^T D^{-1} A$ is cone positive definite. Note that satisfying this assumption simply requires that all of the impedances have some resistive component, which is true of any real power network. In this case, we can bound the condition number of the AC case by the condition number of the real part

of the system and a function of the cone constant. Strictly speaking, our matrices are not cone positive definite because of the 1-dimensional nullspace arising because the nodes are ungrounded. Again, this detail is not significant since our system matrix and our preconditioner have the same nullspace.

LEMMA 22. *If the original system matrix K is cone positive definite, so is the preconditioner M.*

Proof. The edge weights of the support tree are by construction sums of edge weights of the original graph. Therefore, the weighted Laplacian matrix T of the support tree is cone positive definite with the same cone constant as K. The preconditioner M is the Schur complement of T obtained by eliminating the internal tree nodes. Gaussian elimination performed on the internal nodes of the support tree is equivalent to performing series circuit reduction on those nodes [9]. If we reduce a node with edges connecting two nodes in series with admittances c_1 and c_2, the resulting edge has admittance $c_{new} = c_1c_2/(c_1 + c_2)$. Thus, if c_1 and c_2 are in the pointed cone with cone constant μ, so is c_{new} since

$$c_{new} = \frac{c_1c_2}{c_1 + c_2} = \frac{1}{\frac{1}{c_1} + \frac{1}{c_2}}, \tag{12}$$

and pointed cones are closed under addition and under taking reciprocals. Therefore, after eliminating all of the internal nodes of the support tree, we have a graph whose edge weights lie in the original pointed cone. The preconditioner M is the weighted Laplacian matrix of this graph; therefore, M is cone positive definite with the same cone constant μ as the original system matrix K. ∎

THEOREM 23. *In the AC case with impedances lying in a pointed cone in the complex plane, $\kappa(M, K) \leq (1 + \mu^2)\kappa(\mathrm{Re}(M), \mathrm{Re}(K))$, where μ is the cone constant.*

Proof. Assume that $K = A^TD^{-1}A$ as before and that $D^{-1} = E + iF$. Since the preconditioner M is also a weighted Laplacian matrix, we can write $M = R^T\Delta R$ where R is an NAI matrix and Δ is a diagonal matrix. As with the components of K, R is a real matrix made up of 1's, -1's, and 0's, and Δ is a diagonal matrix with complex diagonal entries lying in the same cone as the diagonal entries of D. Let $\Delta = B + iC$.

We first establish bounds on $|\mathbf{x}^*A^TD^{-1}A\mathbf{x}|$ in terms of the real part of D^{-1} and the cone constant μ. Let $\mathbf{w} = A\mathbf{x}$. Let d_j represent the diagonal elements of D^{-1}, e_j represent the diagonal elements of E, and f_j represent the diagonal elements of F. Then we have

$$\begin{aligned}
|\mathbf{x}^*A^TD^{-1}A\mathbf{x}| &= |\mathbf{w}^*D^{-1}\mathbf{w}| \\
&= \left|\sum_j |w_j|^2 \cdot d_j\right| \\
&\leq \sum_j \left(|w_j|^2 \cdot |d_j|\right) \\
&\leq \sum_j \left(|w_j|^2\sqrt{1+\mu^2} \cdot e_j\right) \\
&= |\mathbf{w}^*E\mathbf{w}|\sqrt{1+\mu^2} \\
&= |\mathbf{x}^*A^TEA\mathbf{x}|\sqrt{1+\mu^2}.
\end{aligned} \tag{13}$$

Next we bound $|\mathbf{x}^* A^T D^{-1} A\mathbf{x}|$ from below. In this case we have

$$\begin{aligned} |\mathbf{x}^* A^T D^{-1} A\mathbf{x}| &= |\mathbf{w}^* D^{-1}\mathbf{w}| \\ &= |\mathbf{w}^*(E+iF)\mathbf{w}| \\ &= \left|\sum |w_j|^2 e_j + i\sum |w_j|^2 f_j\right| \\ &\geq \left|\sum |w_j|^2 e_j\right| \\ &= |\mathbf{x}^* A^T E A\mathbf{x}|. \end{aligned} \tag{14}$$

The preconditioner M is also a weighted Laplacian matrix of the form $M = R^T \Delta R$ where R is an NAI matrix and Δ is a diagonal matrix. In addition, by Lemma 22, M is cone positive definite with the same cone constant μ as K. Therefore, the same bounds apply to M. Namely,

$$|\mathbf{x}^* R^T \Delta R\mathbf{x}| \leq |\mathbf{x}^* R^T C R\mathbf{x}|\sqrt{1+\mu^2}, \tag{15}$$

and

$$|\mathbf{x}^* R^T \Delta R\mathbf{x}| \geq |\mathbf{x}^* R^T C R\mathbf{x}|. \tag{16}$$

Using these bounds, we can establish an upper bound on the condition number of the preconditioned AC network in terms of the condition number of the real part of the network.

$$\begin{aligned} \kappa(M,K) &= \max \frac{|\mathbf{x}^* M\mathbf{x}|}{|\mathbf{x}^* K\mathbf{x}|} \cdot \max \frac{|\mathbf{x}^* K\mathbf{x}|}{|\mathbf{x}^* M\mathbf{x}|} \\ &\leq \max \frac{\sqrt{1+\mu^2}|\mathbf{x}^* R^T C R\mathbf{x}|}{|\mathbf{x}^* A^T E A\mathbf{x}|} \cdot \max \frac{\sqrt{1+\mu^2}|\mathbf{x}^* A^T E A\mathbf{x}|}{|\mathbf{x}^* R^T C R\mathbf{x}|} \\ &= (1+\mu^2) \max \frac{|\mathbf{x}^* R^T C R\mathbf{x}|}{|\mathbf{x}^* A^T E A\mathbf{x}|} \cdot \max \frac{|\mathbf{x}^* A^T E A\mathbf{x}|}{|\mathbf{x}^* R^T C R\mathbf{x}|} \\ &= (1+\mu^2)\kappa(\mathrm{Re}(M), \mathrm{Re}(K)). \qquad \square \end{aligned} \tag{17}$$

This analysis seemingly implies that we may as well precondition the AC network based only on its DC components. But our experiments indicate that keeping the imaginary part in the preconditioner gives better results. Therefore, there may exist a stronger analysis of the complex case.

3. Splitting. Since we are assuming that the gap between the magnitudes of high impedance wires and low impedance wires is large, and that removal of the high impedance wires disconnects the graph, the matrix $K = A^T D^{-1} A$ can be arbitrarily ill conditioned. Even though the preconditioner effectively reduces condition number, it does not lead to accurate solution by itself under these conditions. Again letting M be the preconditioner for K, the difficulty is that although the product $M^{-1}K$ is well conditioned, nonetheless the resulting vector $M^{-1}(K\mathbf{v})$ may be computed inaccurately because M^{-1} and K are very ill conditioned separately.

We solve this problem by splitting K into its "large space," i.e., the range of K_1 (where K_1 denotes, as before, the weighted Laplacian of the functioning wires) and its "small space," which is null(K_1). These spaces ("large" versus "small") are reversed for M^{-1}. We can easily identify these spaces from the graph and then split

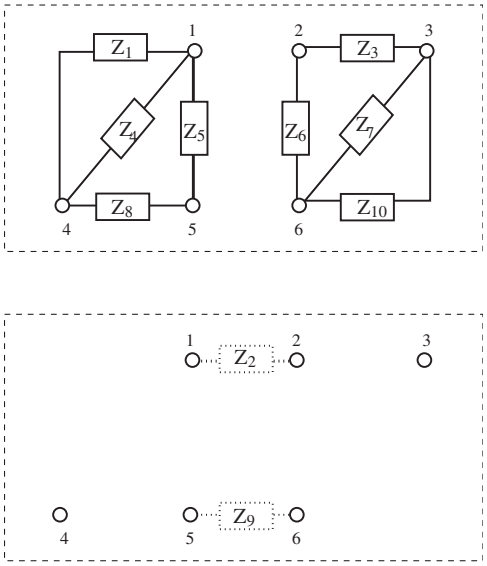


FIG. 7. *The same faulted AC network as shown in Figure* 2 *split into its functioning* (K_1) *and faulted* (K_2) *subgraphs. The top graph shows the functioning subgraphs, and the bottom graph shows the faulted subgraphs.*

$\mathbf{r} = (K_1+K_2)\mathbf{v}$ (in $O(n)$ time) into $\mathbf{r} = \mathbf{r}_R+\mathbf{r}_N$. We then compute $\mathbf{y} = M^{-1}(\mathbf{r}_R+\mathbf{r}_N)$ term by term, changing interim quantities that would be zero in exact arithmetic to zero, then recombining.

The matrix K_1 is just the weighted Laplacian of the functioning part of the system, and K_2 is the weighted Laplacian of the faulted part of the system. The original matrix K then equals $K_1 + K_2$. In Figure 7, we show these two graphs for the same network as shown in Figure 2.

For this network, $K_1 = A_1^T D_1^{-1} A_1$ and $K_2 = A_2^T D_2^{-1} A_2$, where

$$A_1 = \begin{bmatrix} 1 & & & -1 & & & \\ & 1 & -1 & & & & \\ 1 & & & -1 & & & \\ 1 & & & & -1 & & \\ & 1 & & & & -1 & \\ & & 1 & & & -1 & \\ & & & 1 & -1 & & \\ & & 1 & & & -1 & \end{bmatrix},$$

$$D_1 = \operatorname{diag}(\begin{bmatrix} Z_1 & Z_3 & Z_4 & Z_5 & Z_6 & Z_7 & Z_8 & Z_{10} \end{bmatrix}),$$

$$A_2 = \begin{bmatrix} 1 & -1 & & & \\ & & & 1 & -1 \end{bmatrix},$$

$$D_2 = \operatorname{diag}(\begin{bmatrix} Z_2 & Z_9 \end{bmatrix}). \tag{18}$$

In particular, our algorithm is as follows. Each time we form $M^{-1}(K\mathbf{v}) = M^{-1}(K_1 + K_2)\mathbf{v}$, we first split $(K_1 + K_2)\mathbf{v}$ into its subgraphs in range(K_1) and null(K_1). To do this, we form an orthogonal projector PP^T onto the null space of K_1. The projector PP^T is formed based on the graph structure of the network and can be applied in $O(n)$ time as follows. Removing the high impedance (faulty) wires

disconnects the graph into separate subgraphs of functioning wires, which we refer to as K_1-subgraphs. A vector in range(K_1) is defined by the property that its entries over each K_1-subgraph sum to zero. On the other hand, any vector in null(K_1) has the defining property that entries corresponding to nodes within a K_1-subgraph of the graph have the same value. Thus, we build an orthogonal projector onto null(K_1) using a connected subgraph search. Consider forming the product $\mathbf{r} = K\mathbf{v}$. We can split $\mathbf{r}$ according to: $\mathbf{r}_N = PP^TK\mathbf{v}$ and $\mathbf{r}_R = (I - PP^T)K\mathbf{v}$. We compute $\mathbf{r}_N$ as $PP^TK_2\mathbf{v}$ since the other term drops out. Similarly, we compute $\mathbf{r}_R$ as $K_1\mathbf{v} + K_2\mathbf{v} - PP^TK_2\mathbf{v}$. We then invoke our preconditioning algorithm with this split $\mathbf{r}$.

In invoking our preconditioning algorithm, as described in section 2, instead of solving $M\mathbf{y} = \mathbf{r}$, we solve the equivalent (but sparse) problem

$$LU \begin{bmatrix} \mathbf{y} \\ \mathbf{c} \end{bmatrix} = G\mathbf{r}, \tag{19}$$

where LU is the (sparse) LU factorization of T, and G is the matrix that extends a vector with zeros to be the size of the range space of T, i.e.,

$$G\mathbf{r} = \begin{bmatrix} \mathbf{r} \\ \mathbf{0} \end{bmatrix}. \tag{20}$$

We next forward solve for $\mathbf{u}_R = L^{-1}G\mathbf{r}_R$ and $\mathbf{u}_N = L^{-1}G\mathbf{r}_N$. Let S be the set of row numbers of nodes whose paths to the root contain only high-impedance edges. (Because of our definitions, S is exactly the root and children of the root.) Let J be the matrix that zeros out the components in S (i.e., J is a diagonal matrix with zeros in the diagonal positions corresponding to elements of S, and ones elsewhere on the main diagonal). We claim below that in exact arithmetic $\mathbf{u}_R = J\mathbf{u}_R$. Next we do the two backward solves, $\mathbf{w}_N = U^{-1}\mathbf{u}_N$ and $\mathbf{w}_R = U^{-1}J\mathbf{u}_R$. Note the important step in our algorithm of explicitly applying J to $\mathbf{u}_R$ before performing the back substitution.

Finally, we extract and return the part of $\mathbf{w}_R + \mathbf{w}_N$ that corresponds to the $\mathbf{y}$ in $M\mathbf{y} = \mathbf{v}$; i.e., $\mathbf{y} = G^T(\mathbf{w}_R + \mathbf{w}_N)$.

Thus, our algorithm for evaluating $\mathbf{y} = M^{-1}K\mathbf{v}$ is summarized as follows:

$$\begin{aligned} \mathbf{r}_N &= PP^TK_2\mathbf{v}, \\ \mathbf{r}_R &= K_1\mathbf{v} + K_2\mathbf{v} - PP^TK_2\mathbf{v}, \\ \mathbf{u}_N &= L^{-1}G\mathbf{r}_N, \\ \mathbf{u}_R &= L^{-1}G\mathbf{r}_R, \\ \mathbf{w}_N &= U^{-1}\mathbf{u}_N, \\ \mathbf{w}_R &= U^{-1}J\mathbf{u}_R, \\ \mathbf{y} &= G^T(\mathbf{w}_R + \mathbf{w}_N). \end{aligned} \tag{21}$$

Let us now provide three preliminary lemmas about this construction.

LEMMA 24. *Let L and U be lower and upper triangular matrices such that $T = LU$, and let Δ be a diagonal matrix consisting of the main diagonal elements of U. Let $\tilde{U} = \Delta^{-1}U$. Applying L^{-1} sums the entries from the leaves to the root of the support tree. The matrices $\tilde{U}$ and L have the properties that $\|\tilde{U}^{-1}\| \le c_n$, $\|\tilde{U}\| \le d_n$, $\|L^{-1}\| \le k_n$, and $\|L\| \le f_n$, where c_n, d_n, f_n, and k_n are constants that depend only on n.*

Proof. These properties can all be proven combinatorially. We omit the proof. ∎

LEMMA 25. *In exact arithmetic, $J\mathbf{u}_R = \mathbf{u}_R$, where $\mathbf{u}_R$ is defined in* (21).

Proof. Let n_i be a node in the support tree, and let N_i be the set of leaves rooted at n_i. Recall that the leaves of the support tree correspond directly to the nodes of the original network. The weight of the parent edge of n_i is constructed (as in Gremban) to be the sum of the edges connecting the original graph nodes N_i to the rest of the graph. This sum is known as the *frontier* of N_i. Since in our algorithm we force all high impedance edges to be in the top level graph separator, the nodes in the support tree indexed by elements of S are those whose corresponding nodes in the original graph are connected to the rest of the graph by only high impedance edges. This implies that leaves rooted at such a node make up one or more K_1-subgraphs.

Since applying L^{-1} sums the entries from the leaves to the root of the support tree, if $\mathbf{u}_R = L^{-1}G\mathbf{r}_R$, for example, then $u_R(i)$ equals the sum of the entries in $\mathbf{r}_R$ associated with leaves rooted at n_i. As mentioned above, entries in $\mathbf{r}_R$ corresponding to a K_1-subgraph sum to zero since $\mathbf{r}_R \in \text{range}(K_1)$. Therefore, if $i \in S$, then $u_R(i) = 0$ in exact arithmetic. ∎

Let the constants α and β be defined as

$$\alpha = \max\left(\frac{\max_i(D_1^{-1}(i,i))}{\min_i(D_1^{-1}(i,i))}, \frac{\max_i(D_2^{-1}(i,i))}{\min_i(D_2^{-1}(i,i))}\right), \qquad (22)$$
$$\beta = \max_i(D_1^{-1}(i,i)),$$

where the $D_1^{-1}(i,i)$ are the weights of the low impedance edges, and the $D_2^{-1}(i,i)$ are the weights of the high impedance edges.

LEMMA 26. *$\|\Delta^{-1}J\| \le c_n \cdot \alpha/\beta$, where c_n is a constant depending only on n.*

Proof. The elements of Δ are either sums of low impedance weights, sums of high impedance weights, or a mixture of low and high impedance weights. The elements of Δ consisting of the sums of high impedance weights correspond to the row numbers of nodes whose paths to the root contain only high impedance edges, i.e., elements whose associated entries in $\mathbf{r}_R$ correspond to a K_1-subgraph. As shown in Lemma 25, these are the same elements of J that are set to zero. Therefore, the product $\Delta^{-1}J$ is diagonal with the only nonzero elements being sums of low impedance weights. Therefore, $\|\Delta^{-1}J\|$ is equal to the reciprocal of a sum of low impedance (high admittance) weights. The number of weights in the sum is bounded by the number of nodes in the support tree. Therefore, $\|\Delta^{-1}J\| \le c_n \cdot 1/\min_i(D_1^{-1}(i,i)) \le c_n \cdot \alpha/\beta$. ∎

Now we can state our main theorem for this section.

THEOREM 27. *Using the algorithm defined by* (21) *in the presence of roundoff error, the computed $M^{-1}(K\mathbf{v})$ has the form $(M^{-1}K + E)\mathbf{v}$, where E satisfies $\|E\| \le \epsilon_{mach}c_nc_m\alpha + O({\epsilon_{mach}}^2)$, and ϵ_{mach} is machine epsilon.*

Remark 3. In contrast, if we compute $M^{-1}K\mathbf{v}$ in the naive manner, the computed result has the form $(M^{-1}K + E)\mathbf{v}$, where $\|E\| \le \|K\| \cdot \|M^{-1}\| \cdot \epsilon_{mach}$, which could be very large.

Proof. The heart of this proof is in the following two lemmas, which show that $\mathbf{w}_R$ and $\mathbf{w}_N$ are both computed accurately.

LEMMA 28. *The computed $\mathbf{w}_R$ has the form $\hat{\mathbf{w}}_R = [(U^{-1}JL^{-1}G(I - PP^T)(K_1 + K_2)) + E_R]\mathbf{v}$, where $\|E_R\| \le \epsilon_{mach} \cdot c_n \cdot \alpha/\beta \cdot \|K_1 + K_2\| + O({\epsilon_{mach}}^2)$ and α and β are defined by* (22).

Proof. Note that we will reuse the symbol c_n in this proof to mean a constant depending only on n whose value may change from statement to statement.

The range space component of $\mathbf{r}$ is given by $\mathbf{r}_R = K_1\mathbf{v} + K_2\mathbf{v} - PP^TK_2\mathbf{v} = (I - PP^T)(K_1+K_2)\mathbf{v}$, and the computed $\mathbf{r}_R$ has the form $\hat{\mathbf{r}}_R = [(I-PP^T)(K_1+K_2)+E']\mathbf{v}$, where $\|E'\| \le \|K_1\| \cdot \epsilon_{mach} \cdot c_n$. We are assuming here that $\|K_2\| \ll \|K_1\|$.

We next extend $\mathbf{r}_R$ with zeros (i.e., apply G), and perform the forward solve $\mathbf{u}_R = L^{-1}G\mathbf{r}_R$. Note that $\mathbf{u}_R$ is no longer necessarily in range(K_1). The computed results again have the form $\hat{\mathbf{u}}_R = (L^{-1}G + E''')\hat{\mathbf{r}}_R$, where $\|E'''\| \leq \|L^{-1}\| \cdot \epsilon_{mach} \cdot c_n$.

Next, let J be the matrix that sets to zero the elements of $\hat{\mathbf{u}}_R$ that would have been zero in exact arithmetic. (This step is explained in Lemma 25.)

Now we do the back solve $\mathbf{w}_R = U^{-1}J\mathbf{u}_R$. The computed $\mathbf{w}_R$ has the form $(U + F)\hat{\mathbf{w}}_R = J\hat{\mathbf{u}}_R$, where $|F| \leq |U| \cdot \epsilon_{mach} \cdot c_n$ entrywise. Rearranging we get $\hat{\mathbf{w}}_R = (U + F)^{-1}J\hat{\mathbf{u}}_R$. Substituting the Taylor series approximation for $(U + F)^{-1}$ and dropping the high-order terms, we have $\hat{\mathbf{w}}_R = (U^{-1} + U^{-1}FU^{-1})J\hat{\mathbf{u}}_R$.

Recall that $U = \Delta\tilde{U}$, where Δ is a diagonal matrix and $\tilde{U}$ is well conditioned. Since $|F| \leq |U| \cdot \epsilon_{mach} \cdot c_n$ entrywise, F has the same structure, i.e., $F = \Delta\tilde{F}$, where Δ is the same diagonal matrix and $|\tilde{F}| \leq |\tilde{U}| \cdot \epsilon_{mach} \cdot c_n$ entrywise. We can now write the computed $\mathbf{w}_R$ as $\hat{\mathbf{w}}_R = (\tilde{U}^{-1} + \tilde{U}^{-1}\tilde{F}\tilde{U}^{-1})\Delta^{-1}J\hat{\mathbf{u}}_R$.

Putting all of the steps together, we have that

$$\begin{aligned}\hat{\mathbf{w}}_R &= (\tilde{U}^{-1} + \tilde{U}^{-1}\tilde{F}\tilde{U}^{-1})(\Delta^{-1}J)(L^{-1}G + E''') \\ &\qquad \cdot [(I - PP^T)(K_1 + K_2) + E']\mathbf{v} \\ &= [(U^{-1}JL^{-1}G(I - PP^T)(K_1 + K_2)) + E_R]\mathbf{v},\end{aligned} \tag{23}$$

where

$$\begin{aligned}E_R = {} & \tilde{U}^{-1}\tilde{F}\tilde{U}^{-1}\Delta^{-1}JL^{-1}G(I - PP^T)(K_1 + K_2) \\ & + \tilde{U}^{-1}\Delta^{-1}JL^{-1}GE' \\ & + \tilde{U}^{-1}\Delta^{-1}JE'''(I - PP^T)(K_1 + K_2) \\ & + O(\epsilon_{mach}{}^2).\end{aligned} \tag{24}$$

Thus we have

$$\begin{aligned}\|E_R\| &\leq \|\tilde{U}^{-1}\tilde{F}\tilde{U}^{-1}\Delta^{-1}JL^{-1}G(I - PP^T)(K_1 + K_2)\| \\ &\quad + \|\tilde{U}^{-1}\Delta^{-1}JL^{-1}GE'\| \\ &\quad + \|\tilde{U}^{-1}\Delta^{-1}JE'''(I - PP^T)(K_1 + K_2)\| \\ &\quad + O(\epsilon_{mach}{}^2) \\ &\leq \|\tilde{U}^{-1}\tilde{F}\tilde{U}^{-1}\| \cdot \|\Delta^{-1}J\| \cdot \|L^{-1}G(I - PP^T)(K_1 + K_2)\| \\ &\quad + \|\tilde{U}^{-1}\| \cdot \|\Delta^{-1}J\| \cdot \|L^{-1}G\| \cdot \|E'\| \\ &\quad + \|\tilde{U}^{-1}\| \cdot \|\Delta^{-1}J\| \cdot \|E'''\| \cdot \|(I - PP^T)(K_1 + K_2)\| \\ &\quad + O(\epsilon_{mach}{}^2) \\ &\leq \epsilon_{mach} \cdot c_n \cdot \alpha/\beta \cdot \|K_1 + K_2\| + O(\epsilon_{mach}{}^2),\end{aligned} \tag{25}$$

where α and β are as described for Lemma 26. ∎

LEMMA 29. *The computed* $\mathbf{w}_N$ *has the form* $\hat{\mathbf{w}}_N = [(U^{-1}L^{-1}GPP^TK_2) + E_N]\mathbf{v}$, *where* $\|E_N\| \leq \epsilon_{mach}c_n\frac{\alpha}{\beta}\|K_1\| + O(\epsilon_{mach}{}^2)$ *and* α *and* β *are defined by* (22).

Proof. The computed $\mathbf{r}_N$ has the form $\hat{\mathbf{r}}_N = (PP^TK_2 + E^0)\mathbf{v}$, where $\|E^0\| \leq \|K_2\| \cdot \epsilon_{mach} \cdot c_n$, and c_n is a small constant that depends only on n. This follows from the standard properties of matrix-vector multiplication and the fact that $\|PP^T\| = 1$. Note that we will reuse the symbol c_n in this proof to mean a constant depending only on n whose value may change from statement to statement.

We next extend $\mathbf{r}_N$ with zeros (i.e., apply G), and perform the forward solve $\mathbf{u}_N = L^{-1}G\mathbf{r}_N$. Note that $\mathbf{u}_N$ is no longer necessarily in null(K_1).

The computed results again have the form $\hat{\mathbf{u}}_N = (L^{-1}G + E'')\hat{\mathbf{r}}_N$, where $\|E''\| \leq \|L^{-1}\| \cdot \epsilon_{mach} \cdot c_n$.

Now we do the back solve, $\mathbf{w}_N = U^{-1}\mathbf{u}_N$. Similarly to the case for $\mathbf{w}_R$, for $\mathbf{w}_N$ we have

$$\begin{aligned}\hat{\mathbf{w}}_N &= (U^{-1} + U^{-1}FU^{-1})(L^{-1}G + E'')\hat{\mathbf{r}}_N \\ &= (U^{-1} + U^{-1}FU^{-1})(L^{-1}G + E'')(PP^TK_2 + E^0)\mathbf{v} \\ &= (\tilde{U}^{-1} + \tilde{U}^{-1}\tilde{F}\tilde{U}^{-1})\Delta^{-1}(L^{-1}G + E'')(PP^TK_2 + E^0)\mathbf{v} \\ &= [(U^{-1}L^{-1}GPP^TK_2) + E_N]\mathbf{v},\end{aligned} \tag{26}$$

where

$$\begin{aligned}E_N = \tilde{U}^{-1}\Delta^{-1}E''PP^TK_2 &+ \tilde{U}^{-1}\Delta^{-1}L^{-1}GE^0 \\ &+ \tilde{U}^{-1}\tilde{F}\tilde{U}^{-1}\Delta^{-1}L^{-1}GPP^TK_2 \\ &+ O(\epsilon_{mach}{}^2).\end{aligned} \tag{27}$$

So we have

$$\begin{aligned}\|E_N\| \leq \|\tilde{U}^{-1}\| \cdot \|\Delta^{-1}E''PP^TK_2\| &+ \|\tilde{U}^{-1}\| \cdot \|\Delta^{-1}L^{-1}GE^0\| \\ &+ \|\tilde{U}^{-1}\| \cdot \|\tilde{F}\| \cdot \|\tilde{U}^{-1}\| \cdot \|\Delta^{-1}L^{-1}GPP^TK_2\| \\ &+ O(\epsilon_{mach}{}^2).\end{aligned} \tag{28}$$

We have shown in Lemma 24 that $\|\tilde{U}^{-1}\| \leq c_n$ and $\|L^{-1}\| \leq c_n$. As above, $\|E''\| \leq \epsilon_{mach} \cdot c_n\|L^{-1}\|$ and $\|E^0\| \leq \epsilon_{mach} \cdot c_n \cdot \|K_2\|$. Finally, we are assuming that the high impedance weights are significantly larger than the low impedance weights, i.e., that $\|K_2\| \leq \rho\|K_1\|$ for a small constant ρ.

Since Δ is a diagonal matrix with sums of high impedance and low impedance weights on its diagonal, $\|\Delta^{-1}\| \leq 1/\rho \cdot \max_i(1/D_1^{-1}(i,i))$, where $D_1^{-1}(i,i)$ are the weights of the low impedance edges. Therefore, $\|\Delta^{-1}\| \leq \alpha/(\rho\beta)$, where α and β are defined by (22).

Therefore, we can bound the error term $\|E_N\|$ as

$$\begin{aligned}\|E_N\| &\leq c_n\frac{1}{\rho}\frac{\alpha}{\beta}\epsilon_{mach}\rho\|K_1\| + O(\epsilon_{mach}{}^2) \\ &\leq \epsilon_{mach}c_n\frac{\alpha}{\beta}\|K_1\| + O(\epsilon_{mach}{}^2). \qquad \square\end{aligned} \tag{29}$$

Now finally, we conclude the proof of the main theorem. We have shown in Lemmas 28 and 29 that the computed $\mathbf{w}_R$ has the form $\hat{\mathbf{w}}_R = [(U^{-1}JL^{-1}G(I - PP^T)(K_1 + K_2)) + E_R]\mathbf{v}$, where $\|E_R\| \leq \epsilon_{mach} \cdot c_n \cdot \alpha/\beta \cdot \|K_1 + K_2\| + O(\epsilon_{mach}{}^2)$, and the computed $\mathbf{w}_N$ has the form $\hat{\mathbf{w}}_N = [(U^{-1}L^{-1}GPP^TK_2) + E_N]\mathbf{v}$, where $\|E_N\| \leq$

$\epsilon_{mach} c_n \frac{\alpha}{\beta} \|K_1\| + O(\epsilon_{mach}{}^2)$. Therefore, the total computed $\mathbf{w}$ has the form

$$\begin{aligned} \hat{\mathbf{w}} &= \hat{\mathbf{w}}_R + \hat{\mathbf{w}}_N \\ &= [(U^{-1}JL^{-1}G(I - PP^T)(K_1 + K_2)) + E_R]\mathbf{v} \\ &\quad + [(U^{-1}L^{-1}GPP^T(K_1 + K_2)) + E_N]\mathbf{v} \\ &= (M^{-1}K + E)\mathbf{v}, \end{aligned} \tag{30}$$

where

$$\begin{aligned} \|E\| &= \|E_R + E_N\| \\ &\leq \epsilon_{mach} c_n \frac{\alpha}{\beta} \|K\| + O(\epsilon_{mach}{}^2). \end{aligned} \tag{31}$$

Finally, since $\beta = \max_i(D_1^{-1}(i,i))$ and the functioning admittances are assumed to be larger than the faulty admittances, β is the maximum of all of the admittances. In particular, $\|D^{-1}\|/\beta = 1$; therefore, $\|K\|/\beta = c_m$, where c_m is a constant depending on the number of edges in the graph. Therefore, we have

$$\|E\| \leq \epsilon_{mach} c_n c_m \alpha + O(\epsilon_{mach}{}^2). \qquad \square \tag{32}$$

4. Computational experiments. We have tested this algorithm on two qualitatively different graphs: grid-graphs and sample power network graphs from the *MATPOWER* [21] power flow simulation package.

In our analysis, we assume that all of the high-impedance (nonfunctioning) edges are included in the top level of the graph separator. For these examples we defined the high-impedance edges to be exactly those edges cut in the top level graph partition. All remaining edges were taken to be low-impedance (functioning) edges. The low-impedance weights were chosen uniformly at random from a disk in $\mathbf{C}$ centered at 2 of radius 1. The high-impedance (low-admittance) weights were chosen uniformly at random from a disk centered at $2 \cdot 10^{-10}$ of radius $1 \cdot 10^{-10}$. For a 16,384-node system, this choice results in faulted systems of with condition number of approximately 10^{18} (as reported by MATLAB's *condest()* function). We tried several size problems of this kind, with similar results.

All experiments were conducted in MATLAB using sparse matrix operations. The extended preconditioner T was LU-factored without pivoting (and, as mentioned earlier, with no fill-in). The graph partitioning was done using the MATLAB Mesh Partitioning Toolbox [8].

Exact solutions were computed by first forming a random solution vector $\mathbf{x}$ and computing $\mathbf{b} = A\mathbf{x}$. We then used our splitting technique to compute $M^{-1}A^TD^{-1}\mathbf{b}$ accurately to initialize the iterative method.

The results we present in this section are with our algorithm implemented in TFQMR [6]. We have also had success with BiCG [19] and GMRES [16] (not reported here). Note that CG and LSQR are not applicable since (1) is not Hermitian. In addition, we do not use a method such as CSQMR [5] that can take advantage of the complex symmetry of the problem. We have special techniques for computing $M^{-1}(K\mathbf{v})$ accurately and cannot easily reuse our existing methods to compute $K^T(M^{-T}\mathbf{v})$. A transpose method could almost certainly be developed using similar ideas, but we have not tried to implement such a method.

We first compare our algorithm with other iterative methods. In all of the tests that follow, the stopping tolerance is $\|residual\|_2 < (10^{-10} * \|\mathbf{b}\|_2)$.

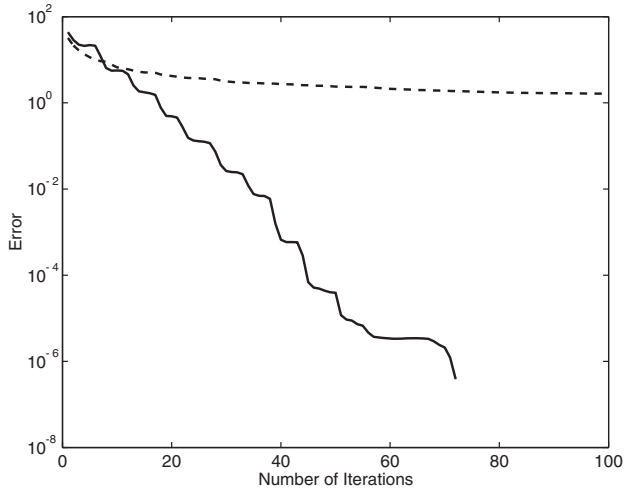


FIG. 8. *Our algorithm and unpreconditioned TFQMR applied to the same faulted* 16,384*-node AC network problem on a grid-graph. The dashed line is unpreconditioned TFQMR without splitting; the solid line is our preconditioned TFQMR with splitting.*

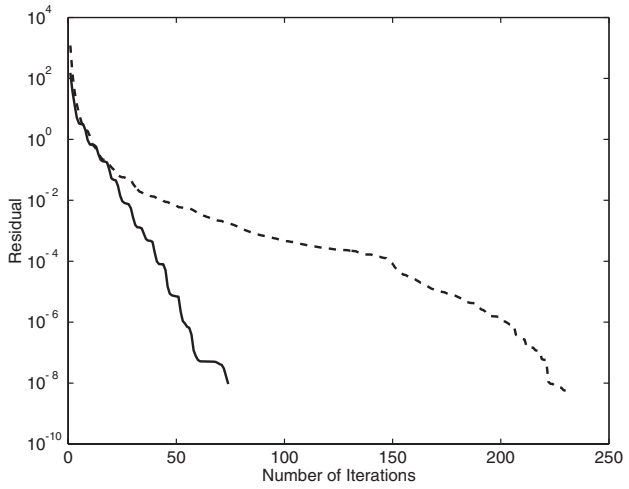


FIG. 9. *Residuals from our algorithm and unpreconditioned TFQMR applied to the same faulted* 16,384*-node AC network problem on a grid-graph. The dashed line is unpreconditioned TFQMR without splitting; the solid line is our preconditioned TFQMR with splitting.*

Figure 8 shows the results from a 16,384-node grid-graph and compares unpreconditioned TFQMR (without splitting) and preconditioned TFQMR with splitting. We see from the figure that our preconditioned TFQMR with splitting is far superior in terms of reducing the error. For TFQMR without preconditioning or splitting, the error was hardly reduced at all from the initial guess. It is important to note that we are plotting error $\|\mathbf{x}^{(j)} - \mathbf{x}\|$ rather than residual $\|K\mathbf{x}^{(j)} - \mathbf{b}\|$. The residual for such an ill-conditioned problem does not give a valid picture of convergence. We can see, for example, in Figure 9 that unpreconditioned TFQMR without splitting does in fact eventually converge (after approximately 250 iterations) in terms of reducing

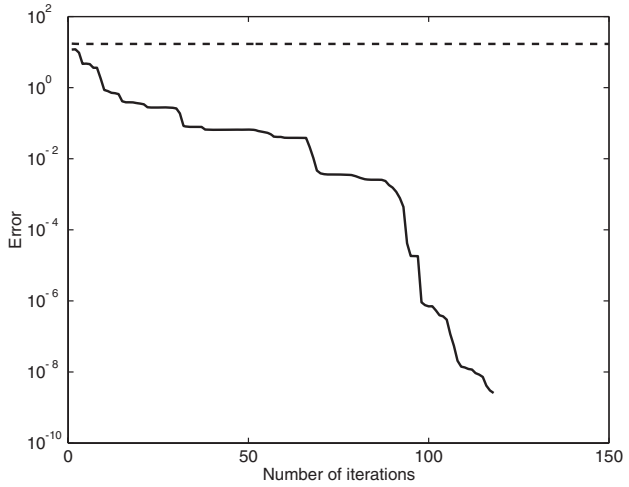


FIG. 10. *Our algorithm and unpreconditioned TFQMR applied to the same* 984*-node AC network problem on a graph from MATPOWER* [21]. *The dashed line is unpreconditioned TFQMR without splitting; the solid line is our preconditioned TFQMR with splitting.*

the updated residual to the specified tolerance, but the error stagnates at approximately 1.2 and never improves. In this example, TFQMR with our preconditioning and splitting took approximately 172.56 seconds, i.e., approximately 2.36 seconds per iteration. All remaining examples in this section will plot the error $\|\mathbf{x}^{(j)} - \mathbf{x}\|$.

Figure 10 shows the results from a 984-node network example from the *MATPOWER* [21] power flow simulation package. As in the previous example, our preconditioned TFQMR with splitting is far superior in terms of reducing the error. For TFQMR without preconditioning or splitting, the error was hardly reduced at all from the initial guess, stagnating at approximately 6.2. In this example, our algorithm took approximately 119.17 seconds (0.99 seconds per iteration) and reduced the error to approximately $2.58 \cdot 10^{-9}$.

Next we compare our algorithm to TFQMR with an incomplete LU (ILU) preconditioner. In Figure 11, we compare TFQMR with an ILU preconditioner (with no fill) to our algorithm on a 16,384-node AC network with no faults. In this example, our algorithm and TFQMR with ILU perform similarly. We use MATLAB's *luinc()* function for the incomplete LU factorization. In this example, unpreconditioned TFQMR without splitting took 180.28 seconds (0.41 seconds per iteration), TFQMR with ILU preconditioning took 75.15 seconds (0.62 seconds per iteration), and our algorithm took 206.84 seconds (2.35 seconds per iteration). In Figure 12, we make the same comparison on a system with faults. Again we see that in a faulted system, our algorithm is far superior in terms of reducing the error. As expected, in unpreconditioned TFQMR without splitting and TFQMR with ILU preconditioning the error stagnates fairly quickly and is never significantly reduced. Our algorithm took 172.56 seconds (2.36 seconds per iteration) and reduced the error substantially.

Next, we examine the effect of the splitting as separate from the support-tree preconditioning. In Figure 13 we compare the results of solving a faulted 16,384-node AC network problem on a grid-graph with and without splitting. This example shows the critical importance of the splitting technique. Although the support-tree preconditioner is a good preconditioner in the sense of improving the condition number

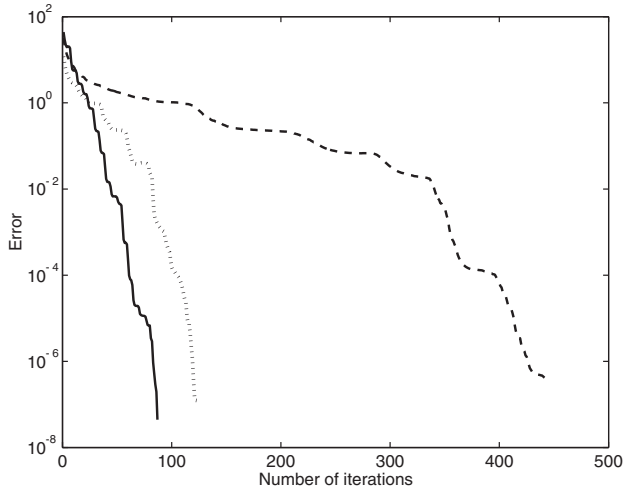


FIG. 11. *Our algorithm, unpreconditioned TFQMR, and TFQMR with ILU preconditioning applied to the same unfaulted* 16,384*-node AC network problem on a grid-graph. The dashed line is unpreconditioned TFQMR without splitting; the dotted line is TFQMR with an ILU preconditioner (no fill); and the solid line is our preconditioned TFQMR with splitting.*

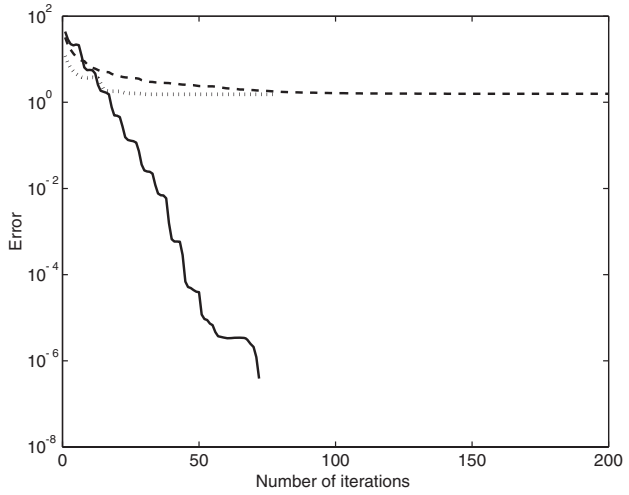


FIG. 12. *Our algorithm, unpreconditioned TFQMR, and TFQMR with ILU preconditioning applied to the same faulted* 16,384*-node AC network problem on a grid-graph. The dashed line is unpreconditioned TFQMR without splitting; the dotted line is TFQMR with an ILU preconditioner (no fill); and the solid line is our preconditioned TFQMR with splitting.*

of the system, it is not helpful unless it can be applied accurately.

Note that we do not perform the opposite experiment of applying our splitting technique without the support-tree preconditioner or with another preconditioner. Our splitting technique is tightly bound to the support-tree preconditioner, and it is not clear how to apply it to an arbitrary preconditioner.

The convergence rate of TFQMR is not known to be related to the condition number in Definition 3 (or indeed, to any condition number). Nonetheless, our experience

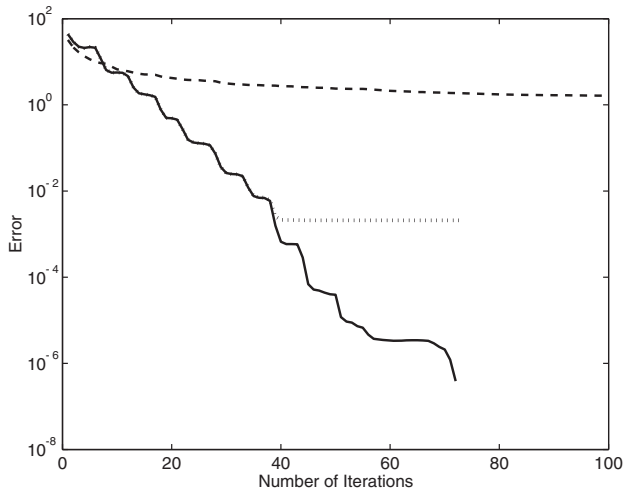


FIG. 13. *Our algorithm, unpreconditioned TFQMR, and TFQMR with our preconditioner but no splitting applied to the same faulted* 16,384*-node AC network problem on a grid-graph. The dashed line shows unpreconditioned TFQMR without splitting; the dotted line is TFQMR with support-tree preconditioning but without splitting; and the solid line is our preconditioned TFQMR with splitting.*

indicates that these two are related. Note that the condition number in Definition 3 is related to the condition number of $(M^*M)^{-1}(K^*K)$, which is the condition number that would apply if we were using CGNR.

Finally, we compare our algorithm with two direct methods, MATLAB's *backslash* command and the complete orthogonal decomposition (COD) method of Hough and Vavasis [13]. On the unfaulted 16,384-node grid-graph problem, MATLAB's *backslash* command is both fast and accurate, taking 9.21 seconds and resulting in a solution with error $2.96 \cdot 10^{-11}$. On the same unfaulted problem, our algorithm takes 203.76 seconds with a resulting error of $4.41 \cdot 10^{-8}$. On the faulted system, however, although *backslash* is still fast (9.23 seconds), the resulting solution has an error of $1.57 \cdot 10^{-3}$. Note that one step of iterative refinement improved this error by approximately one order of magnitude. However further steps of iterative refinement produced no further improvement in the error. In contrast, our algorithm takes 169.42 seconds with a resulting error of $3.87 \cdot 10^{-7}$ on the same faulted system.

The COD algorithm requires SPD systems, so we compare our algorithm with COD using real values for the impedances. Analogously to the complex-symmetric case, low-impedance weights were chosen uniformly at random from an interval on the real line centered at 2 of radius 1. High-impedance weights were chosen from an interval centered at $2 \cdot 10^{-10}$ of radius $1 \cdot 10^{-10}$. The COD algorithm is accurate, even in the presence of faults, but is significantly slower than our algorithm. On an unfaulted 256-node grid system with real edge weights, the COD algorithm took 117.38 seconds with a resulting error of $7.55 \cdot 10^{-15}$, MATLAB's *backslash* command took approximately 0.04 seconds with a resulting error of $2.6 \cdot 10^{-14}$, and our algorithm took approximately 0.94 seconds with a resulting error of $4.64 \cdot 10^{-9}$.

Once we introduce faults, our algorithm and the COD algorithm remain accurate, but the COD algorithm is still significantly slower. On a faulted 1024-node systems with real edge weights, the COD algorithm took 117.45 seconds with a resulting error of $4.94 \cdot 10^{-15}$, MATLAB's *backslash* command took approximately 0.014 seconds with

a resulting error of $5.96 \cdot 10^{-5}$, and our algorithm took approximately 0.84 seconds with a resulting error of $2.23 \cdot 10^{-8}$.

5. Conclusions. We have presented an iterative method for solving complex-symmetric linear systems arising in electrical power networks. We extend Gremban, Miller, and Zagha's [10] support-tree preconditioner to handle the case of faulted AC networks, i.e., complex weights and vastly different admittances. In addition to these extensions, we present a splitting technique that allows us to apply the preconditioner accurately even when the system matrix, and therefore the preconditioner, is arbitrarily ill conditioned. Our computational results show that this iterative method works well in practice in reducing the error.

Note that these results can also apply to the sandstone and shale problem of Vuik [20] and may also apply to interior point methods if we have nullspace information.

REFERENCES

[1] A. R. Bergen, *Power Systems Analysis*, Prentice–Hall, Englewood Cliffs, NJ, 1986.

[2] M. Bern, J. R. Gilbert, B. Hendrickson, N. Nguyen, and S. Toledo, *Support-graph preconditioners*, SIAM J. Matrix Anal. Appl., in review.

[3] E. Y. Bobrovnikova and S. A. Vavasis, *Accurate solution of weighted least squares by iterative methods*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1153–1174.

[4] A. J. Flueck and H.-D. Chiang, *Solving the nonlinear power flow equations with an inexact Newton method using GMRES*, IEEE Trans. Power Systems, 13 (1998), pp. 267–273.

[5] R. W. Freund, *Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 425–448.

[6] R. W. Freund, *A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems*, SIAM J. Sci. Comput., 14 (1993), pp. 470–482.

[7] F. D. Galiana, H. Javidi, and S. McFee, *On the application of a pre-conditioned conjugate gradient algorithm to power network analysis*, IEEE Trans. Power Systems, 9 (1994), pp. 629–636.

[8] J. R. Gilbert, G. L. Miller, and S.-H. Teng, *Geometric mesh partitioning: Implementation and experiments*, SIAM J. Sci. Comput., 19 (1998), pp. 2091–2110.

[9] K. D. Gremban, *Combinatorial Preconditioners for Sparse, Symmetric, Diagonally Dominant Linear Systems*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, 1996.

[10] K. D. Gremban, G. L. Miller, and M. Zagha, *Performance evaluation of a new parallel preconditioner*, in Proceedings of the International Parallel Processing Symposium, IEEE Computer Society, Los Alamitos, CA, 1995, pp. 65–69.

[11] R. Guo and R. D. Skeel, *An algebraic hierarchical basis preconditioner*, Appl. Numer. Math., 9 (1992), pp. 21–32.

[12] E. V. Haynsworth, *Applications of an inequality for the Schur complement*, Proc. Amer. Math. Soc., 24 (1970), pp. 512–516.

[13] P. D. Hough and S. A. Vavasis, *Complete orthogonal decomposition for weighted least squares*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 369–392.

[14] V. E. Howle, *Efficient Iterative Methods for Ill-conditioned Linear and Nonlinear Network Problems*, Ph.D. thesis, Cornell University, Ithaca, NY, 2001.

[15] M. A. Pai, P. W. Sauer, and A. Y. Kulkarni, *A preconditioned iterative solver for dynamic simulation of power systems*, in Proceedings of the IEEE International Symposium on Circuits and Systems, IEEE, New York, 1995, pp. 1279–1282.

[16] 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.

[17] A. Semlyen, *Fundamental concepts of a Krylov subspace power flow methodology*, IEEE Trans. Power Systems, 11 (1996), pp. 1528–1537.

[18] P. M. Vaidya, *Solving linear equations with symmetric diagonally dominant matrices by constructing good preconditioners*, manuscript. A talk based on the manuscript was presented at the IMA Workshop on Graph Theory and Sparse Matrix Computation, Minneapolis, MN, 1991.

[19] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.
[20] C. Vuik, A. Segal, and J. A. Meijerink, *An efficient preconditioned CG method for the solution of a class of layered problems with extreme contrasts in the coefficients*, J. Comput. Phys., 152 (1999), pp. 385–403.
[21] R. Zimmerman and D. Gan, *MATPOWER*, http://www.pserc.cornell.edu/powerweb/.

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 4, pp. 1179–1193

THE SCALING AND SQUARING METHOD FOR THE MATRIX EXPONENTIAL REVISITED*

NICHOLAS J. HIGHAM†

Abstract. The scaling and squaring method is the most widely used method for computing the matrix exponential, not least because it is the method implemented in MATLAB's `expm` function. The method scales the matrix by a power of 2 to reduce the norm to order 1, computes a Padé approximant to the matrix exponential, and then repeatedly squares to undo the effect of the scaling. We give a new backward error analysis of the method (in exact arithmetic) that employs sharp bounds for the truncation errors and leads to an implementation of essentially optimal efficiency. We also give new rounding error analysis that shows the computed Padé approximant of the scaled matrix to be highly accurate. For IEEE double precision arithmetic the best choice of degree of Padé approximant turns out to be 13, rather than the 6 or 8 used by previous authors. Our implementation of the scaling and squaring method always requires at least two fewer matrix multiplications than `expm` when the matrix norm exceeds 1, which can amount to a 37% saving in the number of multiplications, and it is typically more accurate, owing to the fewer required squarings. We also investigate a different scaling and squaring algorithm proposed by Najfeld and Havel that employs a Padé approximation to the function $x\coth(x)$. This method is found to be essentially a variation of the standard one with weaker supporting error analysis.

Key words. matrix function, matrix exponential, Padé approximation, matrix polynomial evaluation, scaling and squaring method, MATLAB, `expm`, backward error analysis, performance profile

AMS subject classification. 65F30

DOI. 10.1137/04061101X

1. Introduction.

The matrix exponential is a much-studied matrix function, owing to its key role in the solution of differential equations. Computation of e^A is required in applications such as nuclear magnetic resonance spectroscopy [8], [18], control theory [5], and Markov chain analysis [20]. Motivated by the applications, mathematicians and engineers have produced a large amount of literature on methods for computing e^A.

A wide variety of methods for computing e^A were analyzed in the classic paper of Moler and Van Loan [16], which was reprinted with an update in [17]. The conclusion of the paper was that there are three or four candidates for best method. One of these, the scaling and squaring method, has become by far the most widely used, not least because it is the method implemented in MATLAB.

In this work we take a fresh look at the scaling and squaring method, giving a sharp analysis of truncation errors and a careful treatment of computational cost. We derive a new implementation that has essentially optimal efficiency and show that it requires at least one less matrix multiplication than existing implementations, including that in MATLAB. Our analysis and implementation are presented in section 2. Section 3 contains a comparison with existing implementations and numerical experiments. The new implementation is found to be typically more accurate than

*Received by the editors July 5, 2004; accepted for publication (in revised form) by I. C. F. Ipsen September 30, 2004; published electronically June 3, 2005. This work was supported by Engineering and Physical Sciences Research Council grant GR/T08739 and by a Royal Society-Wolfson Research Merit Award.

http://www.siam.org/journals/simax/26-4/61101.html

†School of Mathematics, The University of Manchester, Sackville Street, Manchester, England M60 1QD (higham@ma.man.ac.uk, http://www.ma.man.ac.uk/~higham/).

the existing ones, owing to the fact that it usually requires fewer matrix squarings. This work therefore provides another example of the phenomenon, illustrated in the work of Dhillon and Parlett [2], for example, that speed and accuracy are not always conflicting goals in matrix computations.

The standard scaling and squaring method employs Padé approximants to e^x. Najfeld and Havel [18] propose a variation using Padé approximants to the function $x\coth(x)$ instead, and they argue that this approach is more efficient than direct Padé approximation. In section 4 we show that the proposed method is essentially a variation of the standard method, but with weaker supporting error analysis, both in exact arithmetic and in floating point arithmetic.

For other recent work on the scaling and squaring method, concerned particularly with arbitrary precision computations, see Sofroniou and Spaletta [21].

Throughout this paper, $\|\cdot\|$ denotes any subordinate matrix norm. We use the standard model of floating point arithmetic with unit roundoff u [11, sec. 2.2]. Our rounding error bounds are expressed in terms of the constants

$$\gamma_k = \frac{ku}{1-ku}, \qquad \widetilde{\gamma}_k = \frac{cku}{1-cku}, \tag{1.1}$$

where c denotes a small integer constant whose exact value is unimportant.

2. The scaling and squaring method. The scaling and squaring method exploits the relation $e^A = (e^{A/\sigma})^\sigma$, for $A \in \mathbb{C}^{n\times n}$ and $\sigma \in \mathbb{C}$, together with the fact that e^A can be well approximated by a Padé approximant near the origin, that is, for small $\|A\|$. The idea is to choose σ an integral power of 2, $\sigma = 2^s$ say, so that A/σ has norm of order 1, approximate $e^{A/2^s} \approx r_{km}(A/2^s)$, where r_{km} is a $[k/m]$ Padé approximant to the exponential, and then take $e^A \approx r_{km}(A/2^s)^{2^s}$, where the approximation is formed by s repeated squarings. Recall that $r_{km}(x) = p_{km}(x)/q_{km}(x)$ is defined by the properties that p and q are polynomials of degrees at most k and m, respectively, and that $e^x - r_{km}(x) = O(x^{k+m+1})$. The scaling and squaring method method goes back at least to Lawson [15].

The mathematical elegance of the scaling and squaring method is enhanced by the fact that the $[k/m]$ Padé approximants $r_{km}(x) = p_{km}(x)/q_{km}(x)$ to the exponential function are known explicitly for all k and m:

$$p_{km}(x) = \sum_{j=0}^{k} \frac{(k+m-j)!\,k!}{(k+m)!\,(k-j)!}\frac{x^j}{j!}, \quad q_{km}(x) = \sum_{j=0}^{m} \frac{(k+m-j)!\,m!}{(k+m)!\,(m-j)!}\frac{(-x)^j}{j!}. \tag{2.1}$$

Note that $p_{km}(x) = q_{mk}(-x)$, which reflects the property $1/e^x = e^{-x}$ of the exponential function. Later we will exploit the fact that $p_{mm}(x)$ and $q_{mm}(x)$ approximate $e^{x/2}$ and $e^{-x/2}$, respectively, though they do so *much* less accurately than $r_{mm} = p_{mm}/q_{mm}$ approximates e^x. That r_{km} satisfies the definition of Padé approximant is demonstrated by the error expression [6, Thm. 5.5.1]

$$e^x - r_{km}(x) = (-1)^m \frac{k!\,m!}{(k+m)!(k+m+1)!}x^{k+m+1} + O(x^{k+m+2}). \tag{2.2}$$

Diagonal approximants ($k = m$) are preferred, since r_{km} with $k \neq m$ is less accurate than r_{jj}, where $j = \max(k,m)$, but r_{jj} can be evaluated at a matrix argument at the same cost. Moreover, the diagonal approximants have the property that if the eigenvalues of A lie in the open left half-plane then the eigenvalues of $r_{mm}(A)$

have modulus less than 1 (that is, the spectral radius $\rho(r_{mm}(A)) < 1$), which is an important property in applications to differential equations [23, Chap. 8]. We will write the diagonal approximants as $r_m(x) = p_m(x)/q_m(x)$.

Our aim is to choose s, in the initial scaling $A \leftarrow A/2^s$, so that the exponential is computed with backward error bounded by the unit roundoff and with minimal cost. In bounding the backward error we assume exact arithmetic and examine solely the effects of the approximation errors in the Padé approximant.

We begin by considering errors. The choice of s will be based on $\|A\|$, where the norm can be any subordinate matrix norm. Our aim is therefore to bound the backward error in terms of $\|2^{-s}A\|$ and then to determine, for each degree m, the maximum $\|2^{-s}A\|$ for which r_m can be guaranteed to deliver the desired backward error. Moler and Van Loan [16] give a very elegant backward error analysis, from which they obtain a criterion for choosing m; see also Golub and Van Loan [7, sec. 11.3]. Their analysis has two weaknesses. First, it makes an initial assumption that $\|A\| \le 1/2$, whereas, as we will see, there are good reasons for allowing $\|A\|$ to be much larger. Second, it is designed to provide an explicit and easily computable error bound, and the resulting bound is far from being sharp. We now adapt the ideas of Moler and Van Loan in order to obtain a bound that makes no a priori assumption on $\|A\|$ and is as sharp as possible. The tradeoff is that the bound is hard to evaluate, but this is a minor inconvenience because the evaluation need only be done during the design of the algorithm.

Let

$$e^{-A} r_m(A) = I + G = e^H, \tag{2.3}$$

where we assume that $\|G\| < 1$, so that $H = \log(I+G)$ is guaranteed to exist. (Here, log denotes the principal logarithm.) From $\log(I+G) = \sum_{j=1}^{\infty} (-1)^{j+1} G^j/j$, we have

$$\|H\| = \|\log(I+G)\| \le \sum_{j=1}^{\infty} \|G\|^j/j = -\log(1-\|G\|).$$

Now G is clearly a function of A (in the sense of matrix functions [9], [12, Chap. 6]), hence so is H, and therefore H commutes with A. It follows that

$$r_m(A) = e^A e^H = e^{A+H}.$$

Now we replace A by $A/2^s$, where s is a nonnegative integer, and raise both sides of this equation to the power 2^s to obtain

$$r_m(A/2^s)^{2^s} = e^{A+E},$$

where $E = 2^s H$ satisfies

$$\|E\| \le -2^s \log(1-\|G\|)$$

and G satisfies (2.3) with A replaced by $2^{-s}A$. We summarize our findings in the following theorem.

THEOREM 2.1. *Let the diagonal Padé approximant r_m satisfy*

$$e^{-2^{-s}A}\, r_m(2^{-s}A) = I + G, \tag{2.4}$$

where $\|G\| < 1$. *Then*

$$r_m(2^{-s}A)^{2^s} = e^{A+E},$$

where E *commutes with* A *and*

$$\frac{\|E\|}{\|A\|} \le \frac{-\log(1-\|G\|)}{\|2^{-s}A\|}. \qquad \square \tag{2.5}$$

Theorem 2.1 is a backward error result: it interprets the truncation errors in the Padé approximant as equivalent to a perturbation in the original matrix A. (The result holds, in fact, for any rational approximation r_m, as we have not yet used specific properties of a Padé approximant.) The advantage of the backward error viewpoint is that it automatically takes into account the effect of the squaring phase on the error in the Padé approximant and, compared with a forward error bound, avoids the need to consider the conditioning of the problem.

Our task now is to bound the norm of G in (2.4) in terms of $\|2^{-s}A\|$. Define the function

$$\rho(x) = e^{-x}r_m(x) - 1.$$

In view of the Padé approximation property (2.2), ρ has a power series expansion

$$\rho(x) = \sum_{i=2m+1}^{\infty} c_i x^i, \tag{2.6}$$

and this series will converge absolutely for $|x| < \min\{\, |t| : q_m(t) = 0 \,\} =: \nu_m$. Hence

$$\|G\| = \|\rho(2^{-s}A)\| \le \sum_{i=2m+1}^{\infty} |c_i|\theta^i =: f(\theta), \tag{2.7}$$

where $\theta := \|2^{-s}A\| < \nu_m$. It is clear that if A is a general matrix and only $\|A\|$ is known then (2.7) provides the smallest possible bound on $\|G\|$. The corresponding bound of Moler and Van Loan [16, Appx. 1, Lem. 4] is easily seen to be less sharp, and a refined analysis of Dieci and Papini [3, sec. 2], which bounds a different error, is also weaker when adapted to bound $\|G\|$.

Combining (2.7) with (2.5) we have

$$\frac{\|E\|}{\|A\|} \le \frac{-\log(1-f(\theta))}{\theta}. \tag{2.8}$$

Evaluation of $f(\theta)$ in (2.7) would be easy if the coefficients c_i were one-signed, for then we would have $f(\theta) = |\rho(\theta)|$. Experimentally, the c_i are one-signed for some, but not all, m. Using MATLAB's Symbolic Math Toolbox, we have evaluated $f(\theta)$, and hence the bound (2.8), in 250 decimal digit arithmetic, summing the first 150 terms of the series, where the c_i in (2.6) are obtained symbolically. For $m = 1\colon 21$ we have used a zero-finder to determine the largest value of θ, denoted by θ_m, such that the backward error bound (2.8) does not exceed $u = 2^{-53} \approx 1.1 \times 10^{-16}$, the unit roundoff in IEEE double precision arithmetic. The results are shown to two significant figures in Table 2.1.

The second row of the table shows the values of ν_m, and we see that $\theta_m < \nu_m$ in each case, confirming that the bound (2.7) is valid. The inequalities $\theta_m < \nu_m$ also

TABLE 2.1
Maximal values θ_m of $\|2^{-s}A\|$ such that the backward error bound (2.8) does not exceed $u = 2^{-53}$, values of $\nu_m = \min\{|x| : q_m(x) = 0\}$, and upper bound ξ_m for $\|q_m(A)^{-1}\|$.

m	1	2	3	4	5	6	7	8	9	10
θ_m	3.7e-8	5.3e-4	1.5e-2	8.5e-2	2.5e-1	5.4e-1	9.5e-1	1.5e0	2.1e0	2.8e0
ν_m	2.0e0	3.5e0	4.6e0	6.0e0	7.3e0	8.7e0	9.9e0	1.1e1	1.3e1	1.4e1
ξ_m	1.0e0	1.0e0	1.0e0	1.0e0	1.1e0	1.3e0	1.6e0	2.1e0	3.0e0	4.3e0

m	11	12	13	14	15	16	17	18	19	20	21
θ_m	3.6e0	4.5e0	5.4e0	6.3e0	7.3e0	8.4e0	9.4e0	1.1e1	1.2e1	1.3e1	1.4e1
ν_m	1.5e1	1.7e1	1.8e1	1.9e1	2.1e1	2.2e1	2.3e1	2.5e1	2.6e1	2.7e1	2.8e1
ξ_m	6.6e0	1.0e1	1.7e1	3.0e1	5.3e1	9.8e1	1.9e2	3.8e2	8.3e2	2.0e3	6.2e3

confirm the important fact that $q_m(A)$ is nonsingular for $\|A\| \le \theta_m$ (which is in any case implicitly enforced by our analysis).

Next we need to determine the cost of evaluating $r_m(A)$. Because of the relation $q_m(x) = p_m(-x)$ between the numerator and denominator polynomials, an efficient scheme can be based on explicitly computing the even powers of A, forming p_m and q_m, and then solving the matrix equation $q_m r_m = p_m$ [22]. If $p_m(x) = \sum_{i=0}^{m} b_i x^i$, we have, for the even-degree case,

$$\begin{aligned} p_{2m}(A) &= b_{2m}A^{2m} + \cdots + b_2A^2 + b_0I + A(b_{2m-1}A^{2m-2} + \cdots + b_3A^2 + b_1I) \\ &=: U + V, \end{aligned} \tag{2.9}$$

which can be evaluated with $m+1$ matrix multiplications by forming $A^2, A^4, \ldots, A^{2m}$. Then

$$q_{2m}(A) = U - V$$

is available at no extra cost. For odd degrees,

$$\begin{aligned} p_{2m+1}(A) &= A(b_{2m+1}A^{2m} + \cdots + b_3A^2 + b_1I) + b_{2m}A^{2m} + \cdots + b_2A^2 + b_0I \\ &=: U + V, \end{aligned} \tag{2.10}$$

and so p_{2m+1} and $q_{2m+1} = -U + V$ can be evaluated at exactly the same cost as p_{2m} and q_{2m}. However, for $m \ge 12$ this scheme can be improved upon. For example, we can write

$$\begin{aligned} p_{12}(A) &= A^6(b_{12}A^6 + b_{10}A^4 + b_8A^2 + b_6I) + b_4A^4 + b_2A^2 + b_0I \\ &\quad + A\big[A^6(b_{11}A^4 + b_9A^2 + b_7I) + b_5A^4 + b_3A^2 + b_1I\big] \\ &=: U + V, \end{aligned} \tag{2.11}$$

and $q_{12}(A) = U - V$. Thus p_{12} and q_{12} can be evaluated in just six matrix multiplications (for A^2, A^4, A^6, and three additional multiplications). For $m = 13$ an analogous formula holds with the outer multiplication by A transferred to the U term. Similar formulae hold for $m \ge 14$. Table 2.2 summarizes the number of matrix multiplications required to evaluate p_m and q_m, which we denote by π_m, for $m = 1\colon 21$.

The information in Tables 2.1 and 2.2 enables us to determine the optimal algorithm when $\|A\| \ge \theta_{21}$. From Table 2.2, we see that the choice is between $m = 1$, 2, 3, 5, 7, 9, 13, 17, and 21. (There is no reason to use $m = 6$, for example, since the cost of evaluating the more accurate q_7 is the same as the cost of evaluating q_6.) Increasing

TABLE 2.2
Number of matrix multiplications, π_m, required to evaluate $p_m(A)$ and $q_m(A)$, and the measure of overall cost C_m in (2.12).

m	1	2	3	4	5	6	7	8	9	10	
π_m	0	1	2	3	3	4	4	5	5	6	
C_m	25	12	8.1	6.6	5.0	4.9	4.1	4.4	3.9	4.5	
m	11	12	13	14	15	16	17	18	19	20	21
π_m	6	6	6	7	7	7	7	8	8	8	8
C_m	4.2	3.8	3.6	4.3	4.1	3.9	3.8	4.6	4.5	4.3	4.2

from one of these values of m to the next requires an extra matrix multiplication to evaluate r_m, but this is offset by the larger allowed $\theta_m = \|2^{-s}A\|$ if θ_m jumps by more than a factor 2, since decreasing s by 1 saves one multiplication in the final squaring stage. Table 2.1 therefore shows that $m = 13$ is the best choice. Another way to arrive at this conclusion is to observe that the cost of the algorithm in matrix multiplications is, since $s = \lceil \log_2 \|A\|/\theta_m \rceil$ if $\|A\| \geq \theta_m$ and $s = 0$ otherwise,

$$\pi_m + s = \pi_m + \max\left(\lceil \log_2 \|A\| - \log_2 \theta_m \rceil,\, 0\right).$$

(We ignore the required matrix equation solution, which is common to all m.) We wish to determine which m minimizes this quantity. For $\|A\| \geq \theta_m$ we can remove the max and ignore the $\|A\|$ term, which is essentially a constant shift, and so we minimize

$$C_m = \pi_m - \log_2 \theta_m. \tag{2.12}$$

The C_m values are shown in the second line of Table 2.2. Again, $m = 13$ is clearly the best choice. We repeated the computations with $u = 2^{-24} \approx 6.0 \times 10^{-8}$, which is the unit roundoff in IEEE single precision arithmetic, and $u = 2^{-105} \approx 2.5 \times 10^{-32}$, which corresponds to quadruple precision arithmetic; the optimal m are now $m = 7$ and $m = 17$, respectively.

Now we consider the effects of rounding errors on the evaluation of $r_m(A)$. We immediately rule out $m = 1$ and $m = 2$ because r_1 and r_2 can suffer from loss of significance in floating point arithmetic. For example, r_1 requires $\|A\|$ to be of order 10^{-8} after scaling, and then the expression $r_1(A) = (I + A/2)(I - A/2)^{-1}$ loses about half the significant digits in A in double precision arithmetic; yet if the original A has norm of order at least 1 then all the significant digits of some of the elements of A should contribute to the result.

The effect of rounding errors on the evaluation of the numerator and denominator of $r_m(A)$ is described by the following result, which can be proved using techniques from [11].

THEOREM 2.2. *Let $g_m(x) = \sum_{k=0}^m b_k x^k$. The computed polynomial $\widehat{g}_m$ obtained by evaluating g_m at $X \in \mathbb{C}^{n\times n}$ using explicit formation of matrix powers as in the methods above satisfies*

$$|g_m - \widehat{g}_m| \leq \widetilde{\gamma}_{mn}\, \widetilde{g}_m(|X|),$$

where $\widetilde{g}_m(X) = \sum_{i=0}^m |b_k| X^k$. Hence $\|g_m - \widehat{g}_m\|_1 \leq \widetilde{\gamma}_{mn}\, \widetilde{g}_m(\|X\|_1)$. □

Applying the theorem to $p_m(A)$, where $\|A\|_1 \leq \theta_m$, and noting that p_m has all

positive coefficients, we deduce that

$$\begin{aligned}\|p_m(A) - \widehat{p}_m(A)\|_1 &\le \widetilde{\gamma}_{mn}\, p_m(\|A\|_1)\\ &\approx \widetilde{\gamma}_{mn}\, e^{\|A\|_1/2}\\ &\le \widetilde{\gamma}_{mn}\|e^{A/2}\|_1\, e^{\|A\|_1}\\ &\approx \widetilde{\gamma}_{mn}\|p_m(A)\|_1\, e^{\|A\|_1} \le \widetilde{\gamma}_{mn}\|p_m(A)\|_1\, e^{\theta_m}.\end{aligned}$$

Hence the relative error is bounded approximately by $\widetilde{\gamma}_{mn}e^{\theta_m}$, which is a very satisfactory bound, given the values of θ_m in Table 2.1. Replacing A by $-A$ in the latter bound we obtain

$$\|q_m(A) - \widehat{q}_m(A)\|_1 \lesssim \widetilde{\gamma}_{mn}\|q_m(A)\|_1\, e^{\theta_m}.$$

In summary, the errors in the evaluation of p_m and q_m are nicely bounded. This analysis improves that of Ward [24, equation (3.5)], who assumes $\|A\| \le 1$ and obtains absolute error bounds.

To obtain r_m we solve a multiple right-hand side linear system with $q_m(A)$ as coefficient matrix, so to be sure that this system is solved accurately we need to check that $q_m(A)$ is well conditioned. It is possible to obtain a priori bounds for $\|q_m(A)^{-1}\|$ under assumptions such as $\|A\| \le 1/2$ [16, Appx. 1, Lem. 2], $\|A\| \le 1$ [24, Thm. 1], or $q_m(-\|A\|) < 2$ [3, Lem. 2.1], but these assumptions are not satisfied for all the m and $\|A\|$ of interest to us. Therefore we take a similar approach to the way we derived the constants θ_m. With $\|A\| \le \theta_m$ and by writing

$$q_m(A) = e^{-A/2}\big(I + e^{A/2}q_m(A) - I)\big) \equiv e^{-A/2}(I + F),$$

we have, if $\|F\| < 1$,

$$\|q_m(A)^{-1}\| \le \|e^{A/2}\|\,\|(I+F)^{-1}\| \le \frac{e^{\theta_m/2}}{1 - \|F\|}.$$

We can expand $e^{x/2}q_m(x) - 1 = \sum_{i=2}^{\infty} d_i x^i$, from which $\|F\| \le \sum_{i=2}^{\infty} |d_i|\theta_m^i$ follows. Our overall bound is

$$\|q_m(A)^{-1}\| \le \frac{e^{\theta_m/2}}{1 - \sum_{i=2}^{\infty} |d_i|\theta_m^i}.$$

By determining the d_i symbolically and summing the first 150 terms of the sum in 250 decimal digit arithmetic, we obtained the bounds in the last row of Table 2.1, which confirm that q_m is very well conditioned for m up to about 13 when $\|A\| \le \theta_m$.

Our algorithm is as follows. It first checks whether $\|A\| \le \theta_m$ for $m \in \{3, 5, 7, 9, 13\}$ and, if so, evaluates r_m for the smallest such m. Otherwise it uses the scaling and squaring method with $m = 13$.

ALGORITHM 2.3. *This algorithm evaluates the matrix exponential of $A \in \mathbb{C}^{n\times n}$ using the scaling and squaring method. It uses the constants θ_m given in Table* 2.3.

```
% Coefficients of degree 13 Padé approximant.
b(0:13) = [64764752532480000, 32382376266240000, 7771770303897600,
           1187353796428800, 129060195264000, 10559470521600,
           670442572800, 33522128640, 1323241920,
           40840800, 960960, 16380, 182, 1]
```

TABLE 2.3
Constants θ_m needed in Algorithm 2.3.

m	θ_m
3	1.495585217958292e-2
5	2.539398330063230e-1
7	9.504178996162932e-1
9	2.097847961257068e0
13	5.371920351148152e0

6 % Preprocessing to reduce the norm.
7 $A \leftarrow A - \mu I$, where $\mu = \mathrm{trace}(A)/n$.
8 $A \leftarrow D^{-1}AD$, where D is a balancing transformation (or set $D = I$ if balancing does not reduce the 1-norm of A).
9 for $m = [3\ 5\ 7\ 9\ 13]$
10 if $\|A\|_1 \le \theta_m$
11 $X = r_m(A)$ % $r_m(A) = [m/m]$ Padé approximant to A.
12 $X = e^{\mu} DXD^{-1}$ % Undo preprocessing.
13 end
14 end
15 $A \leftarrow A/2^s$ with s a minimal integer such that $\|A/2^s\|_1 \le \theta_{13}$ (i.e., $s = \lceil \log_2(\|A\|_1/\theta_{13}) \rceil$).
16 % Form [13/13] Padé approximant to e^A.
17 $A_2 = A^2$, $A_4 = A_2^2$, $A_6 = A_2A_4$
18 $U = A\big[A_6(b_{13}A_6 + b_{11}A_4 + b_9A_2) + b_7A_6 + b_5A_4 + b_3A_2 + b_1I\big]$
19 $V = A_6(b_{12}A_6 + b_{10}A_4 + b_8A_2) + b_6A_6 + b_4A_4 + b_2A_2 + b_0I$
20 Solve $(-U + V)r_{13} = U + V$ for r_{13}.
21 $X = {r_{13}}^{2^s}$ by repeated squaring.
22 $X = e^{\mu} DXD^{-1}$ % Undo preprocessing.

The cost of Algorithm 2.3 is $\pi_m + \lceil \log_2(\|A\|_1/\theta_m) \rceil$ matrix multiplications, where m is the degree of Padé approximant used, and π_m is tabulated in Table 2.2, plus the solution of one matrix equation.

It is readily checked that the sequences $\theta_{13}^{2k} b_{2k}$ and $\theta_{13}^{2k+1} b_{2k+1}$ are approximately monotonically decreasing with k, and hence the ordering given in Algorithm 2.3 for evaluating U and V takes the terms in approximately increasing order of norm. This ordering is certainly preferable when A has nonnegative elements, and since there cannot be much cancellation in the sums it cannot be a bad ordering [11, Chap. 4].

The Padé approximant r_m at line 11 is intended to be evaluated using (2.10) for $m \le 9$, or as in lines 17–19 for $m = 13$.

The preprocessing in Algorithm 2.3 is precisely that suggested by Ward [24] and attempts to reduce the norm by a shift and a similarity transformation.

The use of the [13/13] Padé approximation in Algorithm 2.3 gives optimal efficiency. However, Table 2.1 reports a bound of 3.0 for $\|q_9(A)^{-1}\|$, which is somewhat smaller than the bound of 17 for $\|q_{13}(A)^{-1}\|$, and C_9 is only slightly larger than C_{13}; therefore the best compromise between numerical stability and efficiency could conceivably be obtained by limiting to maximum degree $m = 9$. We will compare these two degrees experimentally in the next section.

3. Comparison with existing algorithms. We now compare Algorithm 2.3 with existing implementations of the scaling and squaring method that also employ

Padé approximations to e^x.

The function `expm` in MATLAB 7 uses $m = 6$ with $\|2^{-s}A\|_\infty \leq 0.5$ as the scaling criterion and does not employ preprocessing. (`expm` is a built-in function, but `expmdemo1` is an M-file implementation of the same algorithm, and in all our tests `expm` and `expmdemo1` produced exactly the same results.) Sidje [19] uses the same parameters in his function `padm`. Surprisingly, neither `expm` nor `padm` evaluates r_6 optimally: whereas (2.9) requires just 4 multiplications, `expm` uses 5, because it evaluates all the powers $A^2, A^3, \ldots, A^6$, while `padm` expends 7 multiplications in using Horner's method with a variant of (2.9). We note that in `padm`, p_m and q_m are evaluated in increasing order of norms of the terms, as in Algorithm 2.3, whereas `expm` uses the reverse ordering.

Ward [24] uses $m = 8$ with $\|2^{-s}A\|_1 \leq 1$ and carries out the same preprocessing as Algorithm 2.3.

In the following discussion we will assume that all the algorithms use the same norm and ignore the preprocessing.

Since Ward's value of θ is twice that used in `expm` and `padm`, and the [8/8] Padé approximant can be evaluated with just one more matrix multiplication than the [6/6] one, Ward's algorithm would have exactly the same cost as `expm` and `padm` for $\|A\| \geq 0.5$, were the latter algorithms to evaluate r_6 efficiently.

It is clear from our analysis that the three algorithms under discussion are not of optimal efficiency. If the [6/6] or [8/8] Padé approximants are to be used, then one can take larger values of θ, as shown by Table 2.1. Moreover, as we have argued, there is no reason to use the degree 6 or 8 approximants because the degree 7 and 9 approximants have the same cost, respectively.

By considering Tables 2.1 and 2.2 it is easy to see the following:

- When $\|A\|_1 > 1$, Algorithm 2.3 requires one or two fewer matrix multiplications than Ward's implementation, two or three fewer than `expm`, and four or five fewer than `padm`. For example, when $\|A\|_1 \in (2, 2.1)$, the number of matrix multiplications reduces from 8 for `expm` and 7 for Ward's implementation to 5 for Algorithm 2.3 (which takes $m = 9$)—a saving of 37% and 29%, respectively.
- When $\|A\|_1 \leq 1$, Algorithm 2.3 requires no more matrix multiplications than `expm`, `padm`, and Ward's algorithm, and up to 3, 5, and 3 fewer, respectively.

Our analysis shows that all these algorithms have a backward error no larger than u, ignoring roundoff. However, it is well known that rounding errors can significantly affect the scaling and squaring method, because the squaring phase can suffer from severe numerical cancellation. The fundamental problem can be seen in the result [11, sec. 3.5]

$$\|A^2 - fl(A^2)\| \leq \gamma_n \|A\|^2,$$

which shows that the errors in the computed squared matrix are small compared with the square of the norm of the original matrix but not necessarily small compared with the matrix being computed. By using standard error analysis techniques it is possible to derive a forward error bound for the scaling and squaring method, as has been done by Ward [24]. However, with our current knowledge of the e^A problem it is not easy to determine whether a large value for the bound signals potential instability of the method or an ill-conditioned problem.

Since the matrix squarings in the scaling and squaring method are potentially dangerous it seems desirable to minimize the number of them. Algorithm 2.3 uses

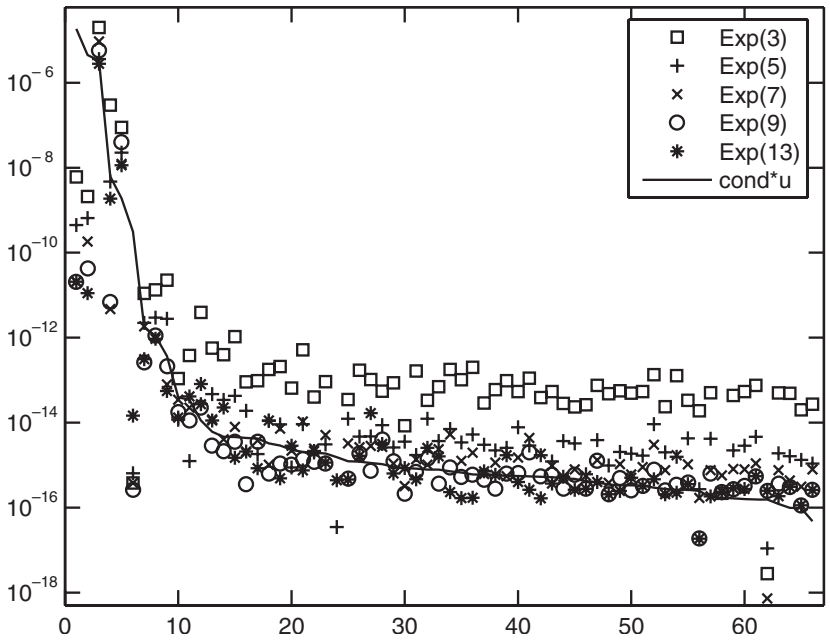


FIG. 3.1. *Normwise relative errors for Algorithm* 2.3 (`Exp(13)`) *and variants with* $m_{\max}$ *restricted to* 3, 5, 7, *and* 9.

one to three fewer squarings than the algorithms with which we have compared it, and hence it has a potential advantage in accuracy.

We now present some numerical experiments, carried out in MATLAB 7.0 (R14), that provide some insight into the accuracy of the scaling and squaring method and of Algorithm 2.3. We took 66 8×8 test matrices: 53 obtained from the function `matrix` in the Matrix Computation Toolbox [10] (which include test matrices from MATLAB itself), together with 13 further test matrices of dimension 2–10 from [1, Ex. 3], [3, Ex. 3.10], [14, Ex. 2 and p. 655], [18, p. 370], and [24, Test Cases 1–4]. We evaluated the relative error in the 1-norm of the computed matrices from `expm`, from Algorithm 2.3, and from a modified version of Algorithm 2.3 in which the maximal degree of the Padé approximant is a parameter, $m_{\max}$. The latter algorithm, denoted by `Exp(`$m_{\max}$`)`, allows us to study the dependence of the error on $m_{\max}$. We did not use any preprocessing in this experiment, although we found that turning on preprocessing in Algorithm 2.3 makes essentially no difference to the results. The "exact" e^A is obtained at 100-digit precision using MATLAB's Symbolic Math Toolbox.

Figure 3.1 compares the errors for the different maximal Padé degrees. It shows a clear trend that the smaller the $m_{\max}$ the larger the error. The solid line is the unit roundoff multiplied by the (relative) condition number

$$\operatorname{cond}(A) = \lim_{\epsilon \to 0} \max_{\|E\|_2 \le \epsilon \|A\|_2} \frac{\|e^{A+E} - e^A\|_2}{\epsilon \|e^A\|_2},$$

which we estimate using the finite-difference power method of Kenney and Laub [13], [9]. For a method to perform in a backward stable, and hence forward stable, manner, its error should lie not far above this line on the graph. In all our figures the results are sorted by decreasing condition number $\operatorname{cond}(A)$. We see that Algorithm 2.3

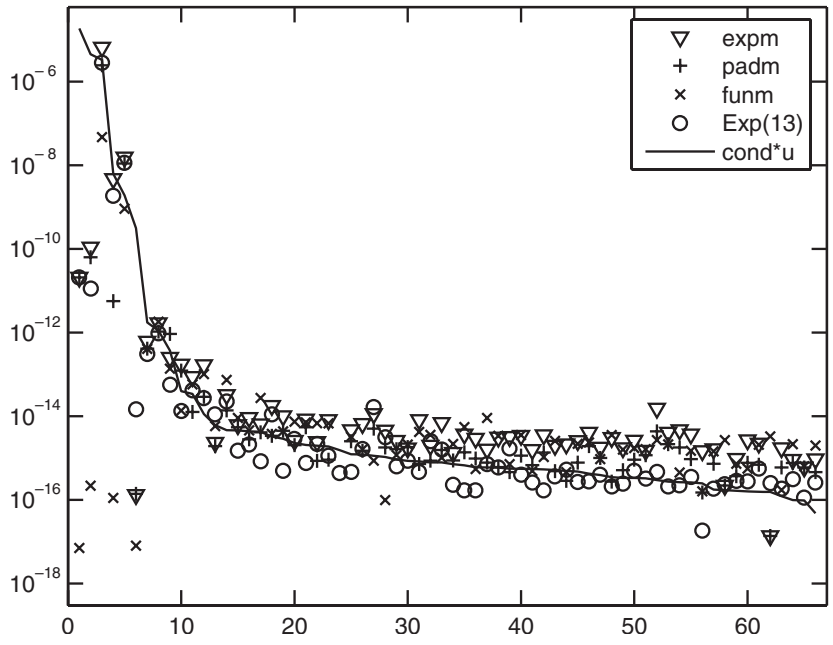


FIG. 3.2. *Normwise relative errors for* `expm`, `padm` (*Sidje*), `funm`, *and Algorithm* 2.3 (`Exp(13)`).

($m_{\max} = 13$) performs in a numerically stable way on this experiment, even though two of the test matrices were chosen to cause the scaling and squaring method to "overscale"—a phenomenon investigated in [3] and [14]. Some instability is apparent for the smaller $m_{\max}$. The numerical results therefore concur with the theory in suggesting that the fewer the number of squarings, the smaller the error.

Figure 3.2 compares Algorithm 2.3 with `expm`, Sidje's function `padm`, and MATLAB 7's `funm`, which implements the Schur–Parlett method of Davies and Higham [1], which is designed for general f. The figure shows that `expm` exhibits minor instability on many of the test matrices.

Finally, Figure 3.3 plots a performance profile [4] for the experiment. Each of the methods is represented by a curve on the plot. For a given α on the x-axis, the y-coordinate of the corresponding point on the curve is the probability that the method in question has an error within a factor α of the smallest error over all the methods, where probabilities are defined over the set of test problems. For $\alpha = 1$, the `Exp(13)` curve is the highest: it intersects the y-axis at $p = 0.52$, which means that this method has the smallest error in 52% of the examples—more often than any other method. For $\alpha \gtrsim 1.6$, `Exp(9)` is more likely than `Exp(13)` to be within a factor α of the smallest error. Since the curve for `expm` lies below all the other curves, `expm` is the least accurate method on this set of test matrices, as measured by the performance profile. Recall that the functions `expm` and `padm` both use $m = 6$ and differ only in how they evaluate r_6, as described at the start of this section.

In interpreting the results it is worth noting that the actual errors the methods produce are sensitive to the details of the arithmetic. The version of Figure 3.3 produced by a prerelease version of MATLAB 7.0 was different, though qualitatively similar. (For example, the `Exp(13)` and `Exp(9)` curves touched at $\alpha = 3$, though they did not cross.)

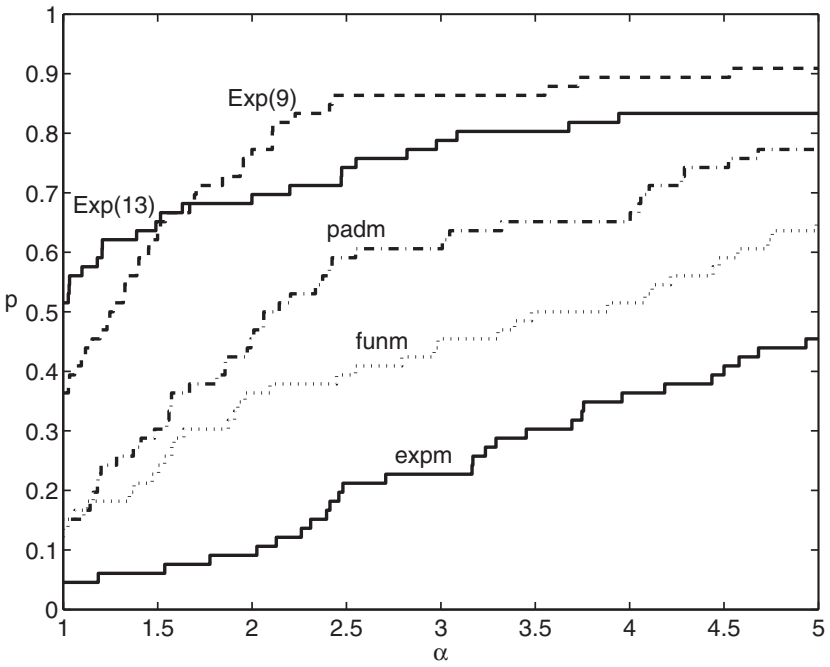


FIG. 3.3. *Performance profile: α is plotted against the probability p that a method has error within a factor α of the smallest error over all methods.*

This experiment shows that in terms of accuracy in floating point arithmetic there is no clear reason to favor `Exp(13)` over `Exp(9)` or vice versa. Our choice of `Exp(13)` in Algorithm 2.3 on the grounds of its lower cost is therefore justified.

4. Indirect Padé approximation. Najfeld and Havel [18, sec. 2] suggest an interesting variation of the standard scaling and squaring method that they claim is more efficient. Instead of approximating the exponential directly, they use a Padé approximation to the even function

$$\begin{aligned}\tau(x) &= x\coth(x) = x(e^{2x}+1)(e^{2x}-1)^{-1}\\ &= 1+\cfrac{x^2}{3+\cfrac{x^2}{5+\cfrac{x^2}{7+\cdots}}},\end{aligned} \tag{4.1}$$

in terms of which the exponential can be written

$$e^{2x} = \frac{\tau(x)+x}{\tau(x)-x}. \tag{4.2}$$

The Padé approximants to τ can be obtained by truncating the continued fraction expansion (4.1). For example, using $\widetilde{r}_{2m}$ to denote the diagonal $[2m/2m]$ Padé approximant to τ,

$$\widetilde{r}_8(x) = \frac{\frac{1}{765765}x^8 + \frac{4}{9945}x^6 + \frac{7}{255}x^4 + \frac{8}{17}x^2 + 1}{\frac{1}{34459425}x^8 + \frac{2}{69615}x^6 + \frac{1}{255}x^4 + \frac{7}{51}x^2 + 1}.$$

The numerators and denominators of $\widetilde{r}_{2m}$ comprise only even powers of x, and so they can be evaluated at a matrix argument A in m matrix multiplications by explicitly forming the required even powers.

The error in r_{2m} has the form

$$\tau(x) - \widetilde{r}_{2m}(x) = \sum_{k=1}^{\infty} d_k x^{4m+2k} = \sum_{k=1}^{\infty} d_k (x^2)^{2m+k}. \tag{4.3}$$

(The error is one order in x higher than the definition of Padé approximant requires, due to the fact that τ is even.) Hence the error in the matrix approximation satisfies

$$\|\tau(A) - \widetilde{r}_{2m}(A)\| \le \sum_{k=1}^{\infty} d_k \|A^2\|^{2m+k} =: \omega_{2m}(\|A^2\|). \tag{4.4}$$

Let θ_{2m} be the largest θ such that $\omega_{2m}(\theta) \le u$. The algorithm of Najfeld and Havel scales $\widetilde{A} \leftarrow A/2^{s+1}$ with $s \ge 0$ chosen so that $\|\widetilde{A}^2\| = \|A^2\|/2^{2s+2} \le \theta_{2m}$. Padé approximation is applied to the scaled matrix, $\widetilde{A}$. The final stage consists of s squarings, just as in the standard scaling and squaring method. Note that there are s squarings rather than $s+1$, because the underlying approximation (4.2) is to e^{2x} and not e^x. Computation of the θ_{2m} and analysis of computational cost in [18] leads Najfeld and Havel to conclude that the choice $m = 8$ of Padé approximant degree leads to the most efficient algorithm.

Detailed study of this algorithm shows that it is competitive in cost with Algorithm 2.3. The following result reveals a close connection with Algorithm 2.3.

THEOREM 4.1. *The* $[2m/2m]$ *Padé approximant* $\widetilde{r}_{2m}(x)$ *to* $x\coth(x)$ *is related to the* $[2m+1/2m+1]$ *Padé approximant* $r_{2m+1}(x)$ *to* e^x *by*

$$r_{2m+1}(x) = \frac{\widetilde{r}_{2m}(x/2) + x/2}{\widetilde{r}_{2m}(x/2) - x/2}.$$

Proof. By (4.3),

$$e_{2m}(x) := \tau(x) - \widetilde{r}_{2m}(x) = O(x^{4m+2}).$$

Then

$$\begin{aligned}
g(x) &:= \frac{\widetilde{r}_{2m}(x) + x}{\widetilde{r}_{2m}(x) - x} = \frac{\tau(x) + x - e_{2m}(x)}{\tau(x) - x - e_{2m}(x)} \\
&= \frac{\tau(x) + x}{\tau(x) - x}\left[\frac{1 - e_{2m}(x)/(\tau(x)+x)}{1 - e_{2m}(x)/(\tau(x)-x)}\right] \\
&= e^{2x}\left[1 - \frac{e_{2m}(x)}{\tau(x)+x} + \frac{e_{2m}(x)}{\tau(x)-x} + O(e_{2m}(x)^2)\right] \\
&= e^{2x}\left[1 + \frac{2x e_{2m}(x)}{(\tau(x)+x)(\tau(x)-x)} + O(e_{2m}(x)^2)\right] \\
&= e^{2x}(1 + xO(e_{2m}(x)) = e^{2x} + O(x^{4m+3}).
\end{aligned}$$

Now $g(x)$ is a rational function with numerator and denominator both of degree at most $2m+1$, and $g(x/2) = e^x + O(x^{4m+3})$. By the uniqueness of Padé approximants to the exponential, $g(x/2) \equiv r_{2m+1}(x)$. ∎

Hence the algorithm of Najfeld and Havel, which takes $m = 8$, is implicitly using the same Padé approximant to e^x as Algorithm 2.3 when the latter takes $m = 9$. The difference is essentially in how A is scaled prior to forming the approximant and in the precise formulae from which the approximant is computed. While the derivation of Najfeld and Havel's algorithm ensures that the error $\|\tau(A) - \widetilde{r}_{2m}(A)\|$ is sufficiently small for the scaled A, what this implies about the error $e^{2A} - (\widetilde{r}_{2m}(A) + A)(\widetilde{r}_{2m}(A) - A)^{-1}$ is unclear, particularly since the matrix $\widetilde{r}_{2m}(A) - A$ that is inverted can be arbitrarily ill conditioned. Moreover, it is unclear how to derive an analogue of Theorem 2.1 that expresses the truncation errors in the Padé approximant to τ as backward errors in the original data.

We conclude that the algorithm suggested by Najfeld and Havel is essentially a variation of the standard scaling and squaring method with direct Padé approximation but with weaker guarantees concerning its behavior both in exact arithmetic (since a backward error result is lacking) and in floating point arithmetic (since a possibly ill-conditioned matrix must be inverted). Without stronger supporting analysis the method cannot therefore be recommended.

5. Conclusions. The scaling and squaring method has long been the most popular method for computing the matrix exponential. By analyzing it afresh we have found that existing implementations of Sidje [19] and Ward [24], and in the function `expm` in MATLAB, are not optimal. While they do guarantee a backward error of order the unit roundoff in the absence of roundoff (that is, solely considering truncation errors in the Padé approximation), they use more matrix multiplications than necessary. By developing an essentially optimal backward error bound for the scaling and squaring method in exact arithmetic that depends on A only through $\|A\|$, we have identified the most efficient choice of degree m of Padé approximation and initial scaling for IEEE double precision arithmetic: $m = 13$, as opposed to $m = 6$ for `expm` and Sidje's algorithm and $m = 8$ for Ward's algorithm, with scaling to ensure $\|A\| \leq 5.4$. A welcome side effect has been to reduce the amount of scaling, and hence the number of squarings in the final stage. This reduction, together with a careful evaluation of the Padé approximation, makes the new algorithm typically more accurate than the old ones (see Figures 3.2 and 3.3).

With the aid of some new error analysis we have shown that all but one part of Algorithm 2.3 is numerically stable. The effect of rounding errors on the final squaring phase remains an open question, but in our experiments the overall algorithm has performed in a numerically stable way throughout.

Acknowledgments. I am grateful to Philip Davies for insightful comments on section 4 and Roy Mathias for suggesting evaluation schemes of the form (2.11).

REFERENCES

[1] 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.
[2] I. S. Dhillon and B. N. Parlett, *Orthogonal eigenvectors and relative gaps*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 858–899.
[3] L. Dieci and A. Papini, *Padé approximation for the exponential of a block triangular matrix*, Linear Algebra Appl., 308 (2000), pp. 183–202.
[4] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213.
[5] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, 3rd ed., Addison-Wesley, Reading, MA, 1998.
[6] W. Gautschi, *Numerical Analysis: An Introduction*, Birkhäuser Boston, Boston, MA, 1997.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[8] T. F. Havel, I. Najfeld, and J. Yang, *Matrix decompositions of two-dimensional nuclear magnetic resonance spectra*, Proc. Natl. Acad. Sci. USA, 91 (1994), pp. 7962–7966.
[9] N. J. Higham, *Functions of a Matrix: Theory and Computation*; book in preparation.
[10] N. J. Higham, *The Matrix Computation Toolbox*, http://www.ma.man.ac.uk/~higham/mctoolbox.
[11] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
[12] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
[13] C. S. Kenney and A. J. Laub, *Condition estimates for matrix functions*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 191–209.
[14] C. S. Kenney and A. J. Laub, *A Schur–Fréchet algorithm for computing the logarithm and exponential of a matrix*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 640–663.
[15] J. D. Lawson, *Generalized Runge-Kutta processes for stable systems with large Lipschitz constants*, SIAM J. Numer. Anal., 4 (1967), pp. 372–380.
[16] C. B. Moler and C. F. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix*, SIAM Rev., 20 (1978), pp. 801–836.
[17] C. B. Moler and C. F. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Rev., 45 (2003), pp. 3–49.
[18] I. Najfeld and T. F. Havel, *Derivatives of the matrix exponential and their computation*, Adv. in Appl. Math., 16 (1995), pp. 321–375.
[19] R. B. Sidje, *Expokit: A software package for computing matrix exponentials*, ACM Trans. Math. Software, 24 (1998), pp. 130–156.
[20] R. B. Sidje and W. J. Stewart, *A numerical study of large sparse matrix exponentials arising in Markov chains*, Comput. Statist. Data Anal., 29 (1999), pp. 345–368.
[21] M. Sofroniou and G. Spaletta, *Efficient matrix polynomial computation and application to the matrix exponential*, talk given at the workshop on "Dynamical Systems on Matrix Manifolds: Numerical Methods and Applications," Bari, Italy, 2004.
[22] C. F. Van Loan, *On the limitation and application of Padé approximation to the matrix exponential*, in Padé and Rational Approximation: Theory and Applications, E. B. Saff and R. S. Varga, eds., Academic Press, New York, 1977, pp. 439–448.
[23] R. S. Varga, *Matrix Iterative Analysis*, 2nd ed., Springer-Verlag, Berlin, 2000.
[24] R. C. Ward, *Numerical computation of the matrix exponential with accuracy estimate*, SIAM J. Numer. Anal., 14 (1977), pp. 600–610.